



TREBALL FINAL DE GRAU



ESCOLA
POLITÈCNICA SUPERIOR
UNIVERSITAT DE LLEIDA
INSPIRING THE FUTURE

Estudiant: Aitor Zaragoza Galiana

Titulació: Grau en Enginyeria Informàtica

Títol de Treball Final de Grau: **Clustering y Analítica de clientes de SEMIC mediante Machine Learning**

Director/a: **Ramón Béjar Torres**

Presentació

Mes: Juliol

Any: 2021

Índice

1	Resumen.....	4
2	Introducción	5
2.1	Motivaciones	5
2.2	Objetivos	6
3	Análisis del problema	8
3.1	SEMIC	8
3.1.1	Organización de las empresas para la venta	9
3.1.2	Verticales del negocio	10
3.1.3	Circuito de ventas con el cliente	11
3.1.4	Organización de los datos	12
3.1.5	¿Qué es un cliente satisfecho?	13
4	Diseño.....	14
4.1	Clientes seleccionados	14
4.2	Diagrama de fases del sistema.....	15
4.2.1	Primera fase	15
4.2.1.1	Datos de partida	16
4.2.1.2	Modelo de datos	19
4.2.2	Segunda fase	19
4.2.2.1	K-Means.....	20
4.2.2.2	Método del codo	20
4.2.3	Tercera fase.....	23
4.2.3.1	Datos añadidos	23
4.2.3.2	Modelo de datos	25
4.3	Creación del grado de satisfacción.....	26
4.4	Ejecución de la herramienta	28
5	Implementación	29
5.1	Qlik Sense	29
5.1.1	Script de carga de datos	29
5.1.1.1	Sintaxis de script.....	30
5.1.2	Requisitos del sistema Qlik Sense	32
5.1.2.1	Plataformas	32
5.1.2.2	Procesadores	32
5.1.2.3	Memoria	32

5.1.2.4	Espacio en el disco.....	32
5.1.2.5	Navegadores compatibles de Microsoft Windows	32
5.1.3	Características del sistema del proyecto.....	33
5.1.4	Conexión a las fuentes de datos	33
5.1.4.1	Base de datos utilizada en el proyecto.....	34
5.1.5	Secciones del script y transformación de los datos	35
5.1.6	Creación de las medidas.....	45
5.2	Python para el desarrollo de la segunda fase	50
5.2.1	Librerías utilizadas.....	50
5.2.1.1	Scikit-learn.....	50
5.2.1.2	Pandas	52
5.2.1.3	Matplotlib.....	52
5.2.2	Tratamiento de los datos	53
5.2.3	Aplicando el método del codo	54
5.2.4	Aplicando el algoritmo de K-Means	55
5.2.5	Elemento central de cada clúster.....	56
5.3	Visualización.....	58
5.3.1	Modificaciones en la carga de datos.....	58
5.3.2	Organización de la visualización.....	59
6	Casos de uso.....	63
6.1	Detección de artículos que generan satisfacción o insatisfacción.....	63
6.2	Provincia de los clientes	65
7	Análisis de Resultados.....	67
8	Bibliografía	69

Tabla de Ilustraciones

Ilustración 1: Estructura de las empresas para la venta	9
Ilustración 2: Diagrama de fases	15
Ilustración 3: Modelo de datos primera fase	19
Ilustración 4: Ejemplos del método del codo	22
Ilustración 5: Modelo de datos tercera fase	25
Ilustración 6: Nivel de desarrollo de los tres últimos años	26
Ilustración 7: Nivel de desarrollo de los Tickets	27
Ilustración 8: Algunas fuentes de datos para crear una conexión en Qlik Sense	33
Ilustración 9: Tabla de mapeo de presupuestos con pedido	37
Ilustración 10: Creación y mapeo de la tabla con los clientes del grupo	38
Ilustración 11: Campos de relación de la tabla de Clientes	39
Ilustración 12: Tratamiento para identificar si el cliente es una persona física.....	40
Ilustración 13: ApplyMap cliente es una empresa del grupo.....	41
Ilustración 14: ApplyMap presupuesto tiene pedido.....	41
Ilustración 15: SQL de carga de las líneas del albarán	42
Ilustración 16: Tabla de Líneas de Albaranes	43
Ilustración 17: Tabla de relación entre Albaranes y Tickets.....	43
Ilustración 18: SQL de carga de los tickets	44
Ilustración 19: Tabla de Tickets	44
Ilustración 20: Ejemplo de Set Expression de Éxito Presupuestos 2018.....	45
Ilustración 21: Medidas de éxito de presupuestos	46
Ilustración 22: Ejemplo de Set Expression de Éxito Presupuestos 2018 con valores nulos tratados	46
Ilustración 23: Medidas de éxito de presupuestos con valores nulos tratados.....	47
Ilustración 24: Set Expression de Importe Albaranes 2018	47
Ilustración 25: Set Expression de Tickets en 2019	48
Ilustración 26: Tabla de datos con medidas anuales	48
Ilustración 27: Fragmento de la tabla lista para exportar de Qlik Sense	49
Ilustración 28: Esquema de algoritmos en scikit-learn	51
Ilustración 29: Fragmento del DataFrame inicial	53
Ilustración 30: DataFrame final al que aplicar el algoritmo	54
Ilustración 31: Implementación método del codo	54
Ilustración 32: Visualización del método del codo.....	55
Ilustración 33: Valor de los centroides de cada clúster	56
Ilustración 34: Filtro y tabla de centroides de los clusters en la visualización.....	60
Ilustración 35: Ejemplo de visualización de los datos de Clientes	60
Ilustración 36: Ejemplo de visualización de los datos de Albaranes	61
Ilustración 37: Ejemplo de visualización de los datos de Tickets	62
Ilustración 38: Artículos consumidos por clientes con mayor nivel de satisfacción.....	63
Ilustración 39: Artículos consumidos por clientes con menor nivel de satisfacción.....	64
Ilustración 40: Provincias de los clientes con mayor nivel de satisfacción	65
Ilustración 41: Provincias de los clientes con menor nivel de satisfacción	65

1 Resumen

En este proyecto, se desarrollará una herramienta capaz de aportar visión a los usuarios de negocio sobre los principales motivos y causas que pueden provocar la insatisfacción en el cliente, así como las características comunes entre los clientes que tienen un menor grado de satisfacción.

En primer lugar, utilizando la plataforma Qlik Sense, se creará un modelo de datos capaz de cagar la información necesaria de las fuentes de datos para generar la métrica de satisfacción del cliente, la cual no tendremos documentada desde un inicio.

En segundo lugar, una vez generada la métrica de satisfacción se procederá a realizar el algoritmo de Machine Learning K-Means para lograr agrupar a todos los clientes en función de su similitud respecto a estas condiciones que se utilicen para definir la satisfacción.

Para finalizar, una vez agrupados los clientes respecto su nivel de satisfacción con la empresa, se procederá a crear las visualizaciones que deberán aportar información sobre las características comunes de los clientes en un mismo grupo.

De esta forma, mediante las visualizaciones el usuario de negocio será capaz de identificar las posibles razones por las que el cliente disminuye la satisfacción con la empresa y podrá llevar a cabo las medidas necesarias para mejorarlo.

2 Introducción

La tendencia de las empresas en generar una mayor cantidad de datos está aumentando de forma exponencial, cada vez se pueden calcular, contabilizar y almacenar más datos de los diferentes procesos dentro de las empresas. Por ello, es necesario extraer información valiosa de estos datos y gracias a ellos, lograr la mayor ventaja competitiva posible.

Este gran volumen de datos es imposible de analizar por una persona por si sola para obtener conclusiones o diseñar predicciones futuras, como modificar los procesos en la venta al cliente o segmentar a clientes para diferentes campañas de marketing.

Además, estos datos no serán de utilidad si no se les ha dado la estructura correcta, ni se ha accedido a los datos necesarios para el análisis requerido. Para ello, existen diferentes plataformas y algoritmos que serán capaces de procesar la información, estructurarla correctamente y compartir el conocimiento que podemos encontrar bajo los datos, el cual deberá ser interpretado por el usuario final de la herramienta.

2.1 Motivaciones

Hace unos meses, empecé a trabajar en una empresa ocupando un puesto dedicado al desarrollo de Business Intelligence corporativo.

Surgió la necesidad de realizar un seguimiento del comportamiento de los clientes tras la venta de productos o servicios, para posteriormente estudiar las causas que provocan una disminución del grado de satisfacción de los mismos y poner solución a este descenso. Por otro lado, otra de las razones sería conocer que productos o servicios producen una mayor satisfacción en el cliente para fomentar su venta.

Esta necesidad surgió porque hasta el momento, no se realizaba ningún análisis de este tipo, y en la actualidad, cada vez es más óptimo el tomar decisiones empresariales basadas en el análisis de datos.

Tras detectar esta necesidad, me pareció una buena idea desarrollar una herramienta que diera la visión sobre este problema a los responsables de cada departamento de la empresa.

Además de poder facilitar el trabajo a la empresa, es un proyecto en el que he podido aplicar y profundizar en conceptos adquiridos en grado de Ingeniería Informática.

2.2 Objetivos

En este trabajo se va a analizar las principales causas y situaciones por las que los clientes reducen su satisfacción con la empresa, y por lo tanto, reducen su participación con esta. Por lo que, el principal objetivo del proyecto será proveer a la empresa de una herramienta que pueda ser capaz de dar visión sobre los principales motivos y causas que pueden provocar la insatisfacción en el cliente.

Basándonos en la métrica que denominaremos satisfacción del cliente, la cual no tendremos documentada desde un inicio, y deberemos establecer el valor de la misma apoyándonos en diferentes campos que puedan reflejar la interacción del cliente con la empresa a lo largo de los últimos tres años.

Para definir la satisfacción del cliente, se va a analizar la evolución de tres campos, estos serán: el importe del cliente en los últimos tres años, el porcentaje de aceptación de presupuestos en los últimos tres años y las incidencias generadas por el cliente en los últimos dos años.

El análisis no se realizará para toda la base de datos de clientes de la empresa, sino que filtraremos a los clientes que cumplan ciertas características, esto se llevará a cabo de esta forma debido a que dependiendo del tipo de cliente, la empresa trabaja con él de una forma u otra.

Una vez definidos los campos que reflejarán la satisfacción del cliente y filtrados los clientes a los que queremos realizar el análisis, nos apoyaremos en técnicas de Machine Learning para lograr agrupar a todos los clientes en función de la similitud respecto a estas condiciones que usamos para definir la satisfacción.

Finalmente, tras obtener la información de a qué grupo de grado de satisfacción pertenece cada cliente, seremos capaces de identificar características comunes entre los clientes insatisfechos, y con esto dar a conocer las posibles causas del deterioro de la relación con el cliente a lo largo del tiempo, las líneas del negocio más afectadas, o incluso los responsables de estas operaciones. Con esta información los responsables de cada departamento pueden hacer una investigación y seguimiento más exhaustivo del problema, pero teniendo la gran ayuda y enfoque inicial sobre la procedencia del mismo.

3 Análisis del problema

Este proyecto se va a desarrollar para la empresa SEMIC y por ello, antes de comenzar debemos conocer cómo se organiza y trabaja la empresa con los clientes, como almacena la información recogida en el proceso de venta y que sería para ellos un cliente satisfecho.

3.1 SEMIC

La empresa Servicios Microinformática S.A. (SEMIC S.A.) es un proveedor global de soluciones y servicios IT con más de 30 años de experiencia en el mercado español ayudando a empresas privadas y a Administraciones Públicas a aprovechar las nuevas tecnologías para mejorar su competitividad ofreciéndoles tecnología, soluciones, servicios y productos que cumplan con sus necesidades.

El grupo SLAKSEMIC está formado por Servicios Microinformática S.A. (SEMIC S.A.), Soluciones TIC para las Organizaciones (SOL TICOR), ESSENTIAM SERVICES S.L. (KONOKONO), SEINSOL S.L., M&C, ACTIUM DIGITAL 2019 S.L., SEMIC Servicios Informáticos S.L. (SEMIC La Seu) y Servicios Microinformáticos Cataluña S.A. (SEMIC Andorra).

3.1.1 Organización de las empresas para la venta

El grupo SLAKSEMIC trabaja de una forma algo peculiar a la hora de vender a los clientes de las respectivas empresas que forman el grupo. Nuestra herramienta de análisis estará centrada en la empresa SEMIC, y como veréis a continuación, conocer la organización de las empresas de grupo para la venta es de vital importancia, ya que debemos realizar diversos tratamientos respecto a esto.

Para entender mejor cómo se organiza el grupo, he creado la siguiente ilustración:

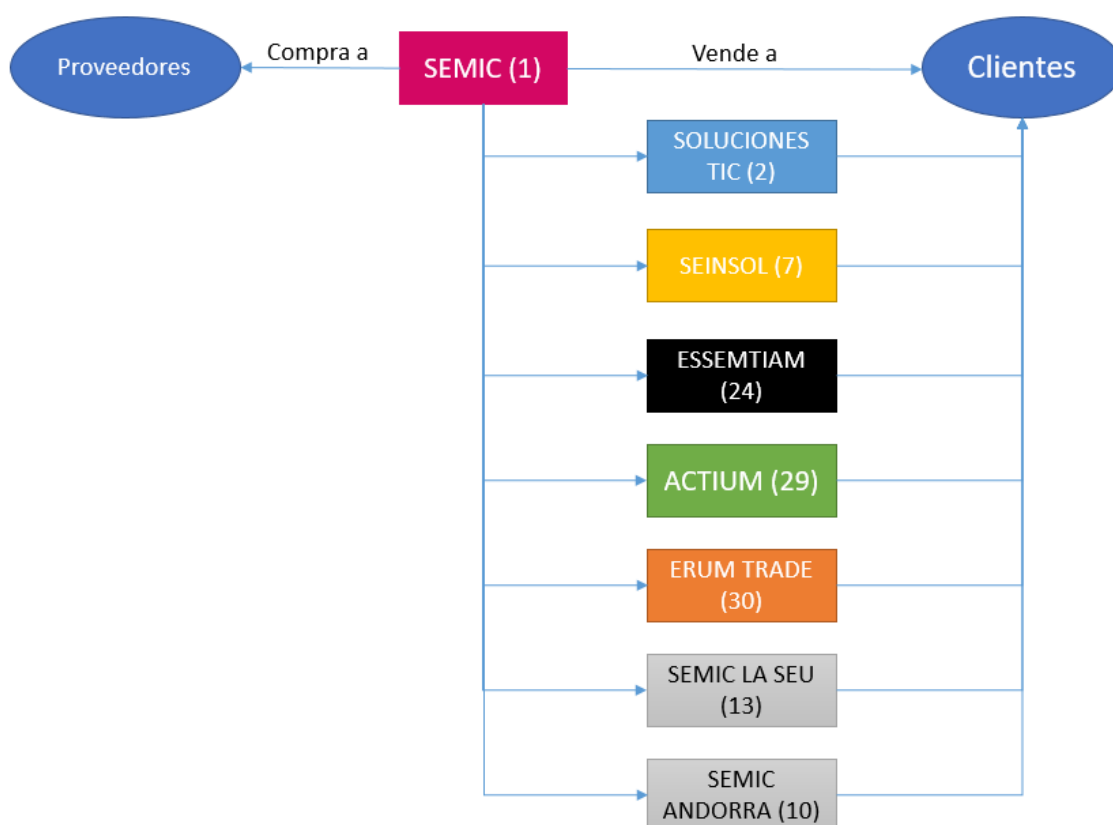


Ilustración 1: Estructura de las empresas para la venta

Como podemos observar en la imagen, la empresa SEMIC es la única que tiene contacto con los proveedores y proporciona los productos necesarios al resto de empresas del grupo, por lo tanto, SEMIC tendrá como clientes a otras empresas y personas físicas que compren sus productos o servicios pero, también aparecerán documentadas como clientes las empresas del propio grupo.

Esta situación es muy importante conocerla ya que encontraremos pedidos de venta entre estas empresas del grupo y, por lo tanto, deberemos ignorar todos sus datos para realizar el análisis. El motivo es que estas ventas no van dirigidas a clientes reales, sino que SEMIC está ejerciendo de “proveedor” al resto de empresas del grupo SLAKSEMIC.

3.1.2 Verticales del negocio

Para comenzar debemos de saber que la vertical es el principal atributo que está directamente relacionado con los clientes y se utilizará para clasificarlos. Este atributo en SEMIC se divide en cuatro posibles grupos que son los siguientes:

- Sanidad: Se suele asignar a clientes que son hospitales, hospitales universitarios, fundaciones dedicadas a recaudar fondos para alguna labor sanitaria, etc.
- Educación: Esta vertical se asigna a clientes como colegios, universidades y en esta parte también entrarían los clientes de los diferentes e-Commerce que SEMIC crea para los colegios.
- Administración Pública: Incluye clientes que son ayuntamientos, consejos comarcales, gobierno, ministerio, juntas, comunidades autónomas, partidos políticos, etc. es decir, cualquier cliente que esté relacionado o se considere como administración pública.
- Indeterminado: En esta vertical encontraremos cualquier tipo de empresa que no trabaje en el ámbito de la sanidad, ni de la educación ni de la administración pública. , suelen ser empresas privadas, autónomos, etc.

Los clientes deben tener al menos una vertical documentada, pero pueden tener más de una. Por ejemplo, los hospitales universitarios pueden pertenecer a la vertical de educación y a la de sanidad.

3.1.3 Circuito de ventas con el cliente

En este apartado veremos el circuito de ventas que tiene SEMIC de forma algo resumida, ya que no entraré en procesos internos que no están directamente relacionados con el cliente. Este circuito de ventas es el más habitual cuando se trata de clientes que pertenecen al atributo de vertical indeterminado.

El circuito comienza desde la captación de posibles clientes, estos pueden generar leads u oportunidades. Un lead se crea cuando la empresa sabe que existe un posible cliente y que este podría tener una necesidad, pero no conoce nada más, es decir, no se tiene claro lo que quiere el cliente. En cambio, una oportunidad es cuando la empresa conoce la necesidad del cliente, incluso sabiendo el artículo y las unidades que necesita, en este caso sí que se tiene bastante claro que es lo que necesita el cliente. Conforme se va conociendo más información de las necesidades del cliente del lead, este termina convirtiéndose en una oportunidad.

Cuando ya tenemos la oportunidad, debemos presentar al cliente uno o varios presupuestos. Si finalmente el cliente está de acuerdo con el presupuesto ofrecido, se genera el pedido de venta hacia el cliente.

Una vez realizadas las gestiones internas necesarias, se termina enviando el pedido al cliente generando en este momento el albarán o los albaranes del pedido que sirven todos los artículos de este.

Finalmente, el cliente ya habrá recibido su pedido y el circuito de ventas queda finalizado. Añadir que el cliente puede generar una incidencia desde el primer momento.

3.1.4 Organización de los datos

Desde todas las diferentes áreas de la empresa se trabaja mediante un ERP. En este se introducen todos los datos para cada documento que se genera en cualquier proceso del negocio, se registran a los clientes y posibles clientes, se generan campañas de marketing para captar nuevos clientes, se registran todas las incidencias, etc. Todos los datos que encontramos se almacenan en diferentes tablas en la base de datos y, por lo tanto, en este trabajo recuperaré todos los datos de ahí.

Los documentos que encontramos dentro del ERP tienen diferentes partes, pero la mayoría de ellos coinciden en una, la cabecera. En las cabeceras podremos encontrar los datos más generales del documento.

Otra parte bastante importante de algunos documentos son las líneas, en estas encontraremos datos más concretos y precisos.

Por ejemplo, en las cabeceras del albarán de ventas encontramos los datos del cliente, la fecha del albarán, el responsable de la operación, la línea de negocio, entre otros. Mientras que en las líneas del albarán de venta, podemos encontrar los artículos de un pedido de venta que hemos entregado al cliente, las unidades de estos, el importe del artículo y el total del albarán, etc.

También es importante recalcar que el ERP es utilizado por todas las empresas del grupo, y por lo tanto, todas las empresas tendrán sus documentos en él. Es importante seleccionar solamente los documentos de la empresa SEMIC.

El número que podemos ver entre paréntesis en la imagen del apartado de Organización de las empresas para la venta, es el número que corresponde a cada empresa dentro del ERP, de esta forma pueden identificar a que empresa pertenece cada documento. Para SEMIC, este campo de empresa contendrá el valor 1.

En las tablas de la base de datos no tendremos creadas las claves primarias de relación como tal, sino que deberemos crear manualmente estas claves de relación concatenando diferentes campos de cada tabla.

3.1.5 ¿Qué es un cliente satisfecho?

Para realizar este proyecto, la empresa ha considerado que un cliente satisfecho deberá de cumplir las siguientes tres condiciones:

Debe ser un cliente que en los últimos tres años ha crecido el éxito en los presupuestos que se le han presentado, es decir, en 2019 ha tenido un porcentaje de aceptación mayor que en 2018 y en 2020 ha tenido una aceptación mayor que en 2019.

Con éxito de los presupuestos nos referimos a que, se ha realizado al menos un pedido del cliente a través del presupuesto que se le ha presentado, ya que un presupuesto con uno o más pedidos relacionados se considerará un presupuesto exitoso.

Por otro lado, también deberá de ascender el gasto que ha realizado en la empresa tras los años, es decir, en 2019 ha gastado más que en 2018 y en 2020 ha gastado más que en 2019.

Podremos acceder al gasto del cliente en SEMIC, a través de los albaranes de venta que tenga el cliente. Si un cliente tiene un albarán, significará que el cliente ha recibido el pedido y se puede dar por cerrada la venta.

Finalmente, también debe ser un cliente que genera menos incidencias o tickets a la empresa, por lo tanto, deben de reducirse sus incidencias de 2019 a 2020.

Estas incidencias se crearán cuando el cliente manda un correo electrónico a SEMIC, mostrando el problema que tiene con su pedido. Podremos acceder al ticket, en el que encontramos la incidencia documentada, a través de los albaranes de venta.

4 Diseño

En esta sección se tratará el diseño de todo el sistema, los diferentes componentes que encontraremos en cada fase y los modelos de datos que obtendremos a partir de las diferentes fuentes de información utilizadas.

4.1 Clientes seleccionados

Para ser capaces de desarrollar el sistema correctamente debemos de tener claro que clientes son los que debemos procesar, ya que dependiendo del tipo de cliente que seleccionemos se debe de tratar su información de una forma u otra.

Así que, en este apartado se mostrarán las características que deben cumplir los clientes para que se tengan en cuenta en el análisis del proyecto.

La principal característica que deben cumplir es que deben pertenecer a la vertical de indeterminado, la cual hemos explicado en el apartado de verticales del negocio.

También, el cliente debe de ser una empresa y no una persona física o autónomo. Para ser capaces de distinguir esta característica se utilizará el N.I.F. que tengan documentado en el ERP. Si este empieza con un carácter igual a X o Y, o con un número podremos afirmar que es una persona física y, por lo tanto, excluirlo del análisis.

Además, solo se seleccionarán los clientes documentados en la empresa 1, es decir, en SEMIC. Y estos clientes no deberán tener fecha de baja, o en el caso de que la tengan, deberá ser mayor a la fecha final del análisis. Esta fecha final del análisis será el año anterior al año en el que se ejecute la herramienta.

Finalmente, debemos excluir a los clientes de SEMIC que pertenezcan al mismo grupo de empresas, ya que estos no serán clientes reales.

4.2 Diagrama de fases del sistema

En este apartado, vamos a analizar las fases por las que deberá pasar nuestra herramienta para lograr obtener los resultados esperados, siendo capaz de aportar valor a los usuarios finales sobre los motivos o causas del deterioro de la relación entre la empresa y el cliente.

Para poder dar una mejor visión sobre las tres principales fases de las que se compone la herramienta se ha creado la siguiente ilustración:

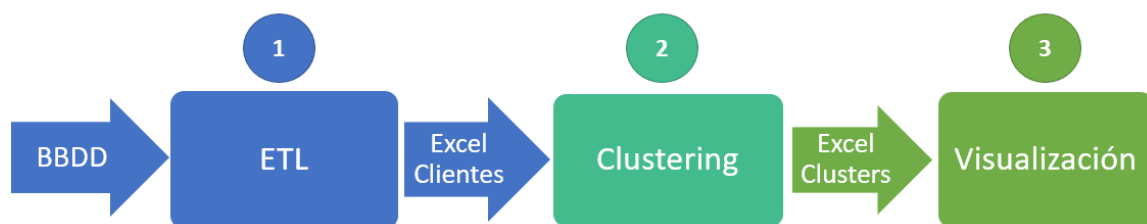


Ilustración 2: Diagrama de fases

4.2.1 Primera fase

En la primera fase se deberá recuperar los datos necesarios de la base de datos de la empresa, para después realizar ciertos tratamientos de estos y establecer las relaciones suficientes entre tablas para crear un modelo de datos, mediante el cual podamos obtener correctamente la información necesaria para identificar la satisfacción de cada cliente.

Finalmente, el resultado de la primera fase será una visualización en formato de tabla que contendrá toda la información organizada por clientes, en la que se habrán creado tres campos que nos aportarán información sobre la evolución del importe, de los presupuestos y de los tickets de los clientes. Esta tabla la exportaremos en un archivo Excel el cual utilizaremos de entrada para la segunda fase.

4.2.1.1 Datos de partida

Para realizar el proyecto tendremos acceso a toda la información que podemos encontrar en el ERP, toda esta información estará organizada mediante tablas en la base de datos de la empresa.

Como hemos comentado anteriormente el proyecto se desarrolla en torno a la satisfacción del cliente. Para poder generar los campos que representarán esta medida necesitamos acceder a las cabeceras y las líneas de los albaranes, a las cabeceras de los presupuestos, y para terminar a la tabla general de tickets. Pero para poder acceder a los tickets relacionados con los albaranes entregados al cliente deberemos utilizar una tabla intermedia de relación entre estos dos documentos.

Finalmente, necesitaremos acceder a la tabla de clientes, la cual será la tabla central a la que se relacionarán la mayoría de las tablas nombradas anteriormente. Para poder filtrar correctamente a los clientes necesitaremos acceder a la tabla de atributos relacionada con el cliente.

A continuación, se mostrará información sobre los campos de las tablas que necesitaremos para el inicio del proyecto:

- Tabla de clientes:

La tabla en la base de datos se denomina *clien*, los datos que serán necesarios de esta tabla serán el código del cliente y el N.I.F del cliente, el cual utilizaremos para comprobar si el cliente es una persona física o es una empresa. El campo de código de cliente será único para cada cliente de la empresa SEMIC y, por lo tanto, la clave primaria de relación con los clientes será mediante los campos de empresa y código de cliente (*cod_cli*).

- Tabla de las cabeceras de los albaranes de ventas:

La tabla en la base de datos se denomina *albcab*, en esta tabla encontraremos la fecha del albarán, el código del cliente al que pertenece el albarán, el cual será utilizado para excluir a los clientes que pertenecen al grupo y crear la relación con el resto de clientes. En los albaranes tendremos el cliente documentado, entonces

podremos trazar la relación hacia el cliente mediante el campo empresa y código cliente.

También, comentar que los campos que identificarán a un albarán, serán los campos de empresa, código de la serie del albarán y número del albarán.

- Tabla de las líneas de los albaranes de ventas:

La tabla en la base de datos se denomina *alblin*. La información principal que debemos recuperar de esta tabla será el importe de todas las líneas de los albaranes. Esta tabla tiene una fácil trazabilidad con la tabla de cabeceras de los albaranes de ventas, ya que tiene documentados los campos de empresa, código de la serie del albarán y número del albarán del que pertenece cada línea de la tabla *alblin*.

- Tabla de las cabeceras de los presupuestos:

La tabla en la base de datos se denomina *precab*. Los datos necesarios de esta tabla serán la fecha del presupuesto, también el código del cliente que, junto con el campo de empresa, deberemos utilizar como relación con la tabla de Clientes. Además, deberemos realizar el mismo tratamiento que en las cabeceras del albarán para excluir las empresas del grupo.

- Tabla de tickets:

La tabla en la base de datos se denomina *ticket*. Los datos necesarios de esta tabla serán la fecha de alta de la incidencia y el identificador de la incidencia. Como he comentado antes, en esta tabla no podemos encontrar una trazabilidad directa hacia los albaranes, así que ira relacionada a una tabla intermedia. La clave primaria con la que se relaciona estará compuesta por los campos empresa, código de la serie del ticket y número del ticket.

Finalmente, comentar que podremos contar el número de tickets que tenemos mediante al campo de número de ticket, que será único para cada ticket.

- Tabla intermedia entre Albaranes y Tickets:

La tabla en la base de datos se denomina *tckdoc*. Como ya se ha comentado, las tablas de cabeceras de albaranes de venta y la de tickets no se pueden relacionar entre sí, por lo tanto, necesitaremos esta tabla intermedia que tiene disponibles los campos de identificador del albarán e identificador del ticket.

En definitiva, esta tabla tiene la capacidad de, a través de un identificador de albarán de venta, devolvernos todos los identificadores de tickets que están relacionados con ese albarán.

- Tabla de las cabeceras de los pedidos de ventas:

En esta tabla, podremos encontrar toda la información sobre las cabeceras de los pedidos de ventas. En la primera parte del proyecto, la utilizaremos para ser capaces de conocer si un presupuesto ha tenido un pedido o no, ya que si se ha realizado un pedido significara que ha sido un presupuesto exitoso.

Esto lo sabremos debido a que en las cabeceras del pedido de ventas, tenemos disponibles dos campos que hacen referencia al presupuesto del que se genera el pedido. Si los encontramos documentados con un número de presupuesto, significará que se ha llevado a cabo el pedido de ese presupuesto, sino tendrán valor nulo.

4.2.1.2 Modelo de datos

Al analizar los datos de partida con la que la herramienta iniciará la primera fase y las relaciones que se deberán establecer entre las tablas nombradas, el modelo de datos resultante será el siguiente:

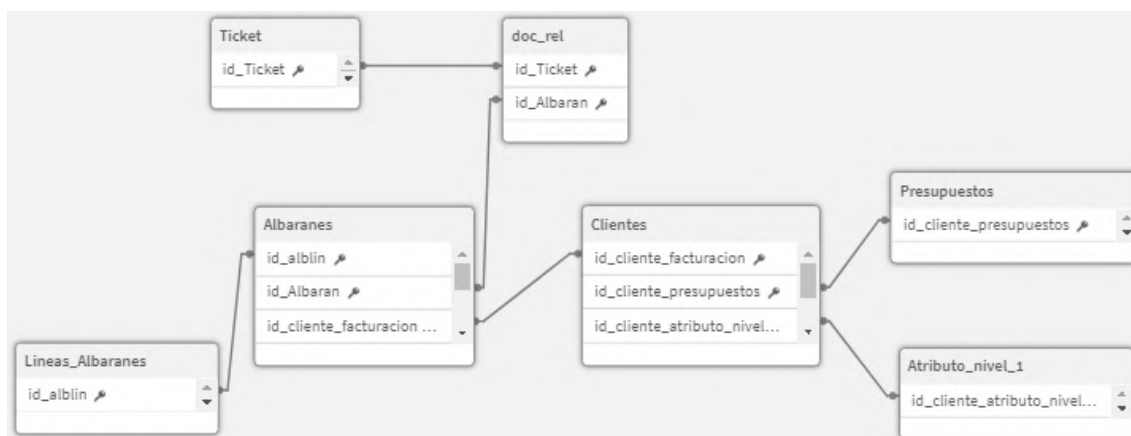


Ilustración 3: Modelo de datos primera fase

4.2.2 Segunda fase

En la segunda fase, mediante el algoritmo de clustering K-Means se deberá obtener el grupo de satisfacción al que pertenece cada cliente. Para ello, utilizaremos como datos de entrada las tres columnas que muestran la evolución de los clientes, las cuales las obtenemos del Excel resultante en la primera fase. También se deberá realizar algún tratamiento del Excel de entrada, para eliminar a los clientes que no están activos y seleccionar únicamente las tres columnas que serán la entrada del algoritmo K-Means.

Finalmente, el resultado de la segunda fase será un nuevo archivo Excel, el cual contendrá a los clientes activos del archivo inicial acompañados del clúster al que pertenecen.

4.2.2.1 K-Means

El algoritmo K-Means, que fue creado por MacQueen en 1967, es el algoritmo de clustering más conocido y utilizado ya que su aplicación es muy simple y eficaz. Sigue un simple procedimiento de clasificación de un conjunto de objetos en un determinado número K de clústeres, estos K clústeres se deben determinar a priori.

El nombre de K-Means es debido a que representa cada uno de los clúster por la media de sus puntos, es decir, por su centroide. Por lo tanto, cada clúster es caracterizado por el punto central de los elementos que lo componen.

El algoritmo del K-Means se realiza en 4 etapas:

- **Etapas 1:** Primero se elige el número de K clusters en el que queremos dividir nuestros datos. Tras esto, el algoritmo elige aleatoriamente K puntos que serán los baricentros y formarán los K clúster iniciales.
- **Etapas 2:** Reasigna cada elemento del clúster a su baricentro más cercano según una medida de distancia, por defecto será la distancia euclídea.
- **Etapas 3:** Una vez que todos los objetos están reasignados en su clúster, se tiene que recalcular y asignar los nuevos centros de los K clústers.
- **Etapas 4:** Repetir las etapas 2 y 3 hasta que no se hagan más reasignaciones. Esto querrá decir que todos los elementos están asignados correctamente en sus respectivos clústers.

En unas pocas iteraciones el algoritmo puede lograr agrupaciones que logren ser óptimas.

4.2.2.2 Método del codo

Como se ha comentado, el primer paso del algoritmo de K-Means es elegir el número de K clúster en el que queremos dividir nuestros datos. Para ello, no existe un método absoluto, pero sí que se pueden realizar diferentes técnicas que nos ayudan a elegir un número apropiado de clusters, como el método del codo.

Este método, utiliza los valores de la inercia obtenidos tras aplicar repetidamente el algoritmo de K-Means para diferentes números de clusters. La inercia, es la suma de las distancias al cuadrado de cada objeto del clúster a su centroide: [1]

$$Inercia = \sum_{i=0}^N |x_i - y_i|^2$$

Una vez obtenidos todos los valores de la inercia, tras aplicar el algoritmo para los valores de 1 a N clusters, se debe representar estos resultados en una gráfica lineal. En esta podremos ver cómo se desarrolla el valor de la inercia respecto al número de clusters.

El objetivo de esta visualización es ser capaces de ver en qué punto de la gráfica, es decir en qué valor de K clusters, se aprecia un cambio en la evolución de la inercia. Observaremos una línea que representa una forma similar a la de un brazo y su codo, el punto en el que apreciamos ese “codo”, será en el que bajará la intensidad de la disminución de la inercia. Este punto es el que debemos seleccionar como valor de K para el número óptimo de clusters.

En este proyecto, el método del codo deberá aplicarse de forma visual. Así que, deberá ser el usuario el que identifique el número óptimo de clusters y el que introduzca el valor al algoritmo de K-Means para que este realice una óptima agrupación.

A continuación se muestran algunos ejemplos de gráficos donde se representa, a la izquierda, el método del codo y, a la derecha, los clusters seleccionados utilizando esta técnica.

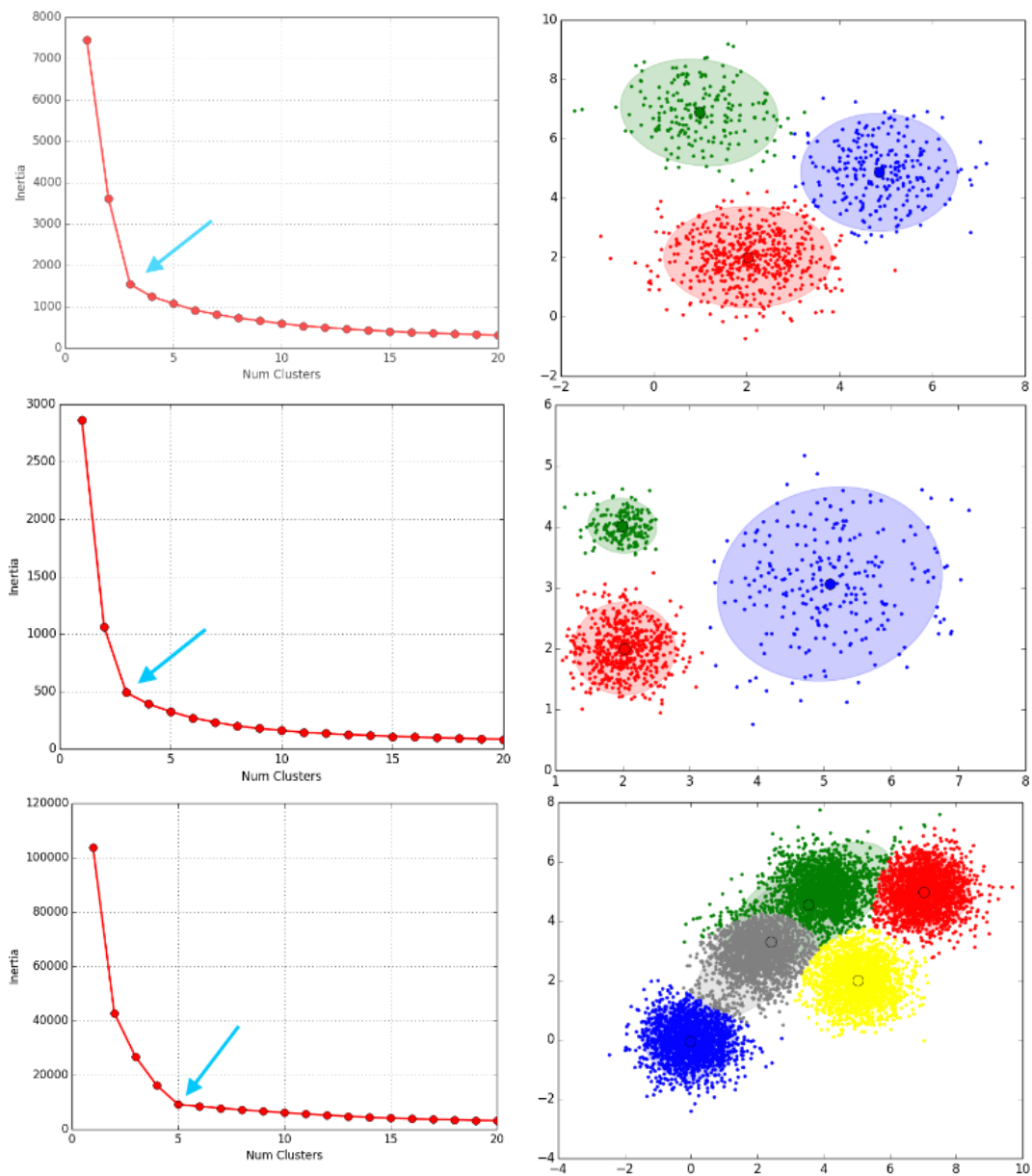


Ilustración 4: Ejemplos del método del codo

Como se puede apreciar en los gráficos y en la nube de puntos que representa los datos con los que se trabajan en los ejemplos, el método del codo selecciona el número de clusters de forma bastante coherente y se ajusta a los resultados esperados.

4.2.3 Tercera fase

En la tercera y última fase, deberemos actualizar el modelo de datos utilizado en la primera fase. A este modelo se deberá añadir la carga del Excel resultante de la segunda fase y ciertas tablas de la base de datos, las cuales nos aportarán una información más precisa sobre las características comunes entre los clientes que pertenecen a un mismo clúster. El resultado de esta fase será lograr aportar valor mediante visualizaciones al usuario de negocio final, para que pueda identificar problemas e incongruencias si las hay.

4.2.3.1 Datos añadidos

Como se ha comentado, en la tercera fase se deben añadir nuevas tablas de la base de datos al modelo de datos resultante en la primera fase. El objetivo de la última fase será dar a conocer las características comunes de los clientes que disminuyen o aumentan su satisfacción con la empresa, por lo que se deberá agregar la tabla de artículos, la cual nos aportará información sobre los productos o servicios que consumen los clientes. También, se agregarán nuevos campos a las tablas ya cargadas en la primera fase para conocer más en detalle las características de estos documentos, como por ejemplo la línea de negocio o el responsable de operación.

A continuación se muestra información sobre las tablas añadidas y alguno de sus campos:

- Tabla de clusters de clientes:

Esta tabla se creará a partir del archivo Excel generado al final de la segunda fase. Estará compuesta de dos columnas, las cuales serán el código del cliente y el clúster al que pertenece. También crearemos un campo de empresa forzándolo siempre con valor igual a 1, ya que será la empresa SEMIC, para poder relacionarlo correctamente con la tabla de clientes.

- Tabla de los puntos centrales de cada clúster

Se ha añadido esta tabla para que el usuario sea capaz de identificar fácilmente que clúster pertenece a los clientes con características cercanas a la satisfacción o a la insatisfacción, basándose en el valor de las tres medidas en el punto central de cada clúster.

Estará compuesta por cuatro columnas las cuales serán el identificador del número de clúster, el nivel medio de éxito de presupuestos, el nivel medio de importe de albaranes y el nivel medio de tickets. Esta tabla se relacionará con la tabla de clúster de clientes mediante el campo clúster, que será el número identificador del clúster.

- Tabla de artículos:

En la tabla de artículos encontraremos información correspondiente a todos los artículos disponibles en las empresas. Algunos de estos datos serán la línea de negocio a la que pertenece el producto, el fabricante, el proveedor, si es un equipo a medida, etc. Un artículo será un equipo a medida cuando se modifique su hardware antes de ser vendido al cliente para que el equipo aumente sus prestaciones.

También, tendremos disponible la empresa y código de artículo que utilizaremos para establecer la relación con las líneas de los documentos. Finalmente, como comprender de que artículo se trata mediante su código de artículo es algo difícil, podremos utilizar un campo de descripción del artículo.

- Nuevos campos añadidos a las tablas existentes:

A las cabeceras de los presupuestos, pedidos y albaranes se les han añadido los campos de línea de negocio, responsable de la operación, agente asignado, etc. En las cabeceras del pedido también se ha añadido la forma de pago para ver si tiene alguna relación con los clientes de un mismo grupo.

Además, ahora tendremos disponibles las líneas de los presupuestos, pedidos y albaranes. De esta forma, podremos acceder a todos los artículos que contienen estos documentos, así como sus unidades, precio, precio de coste, entre otros.

4.2.3.2 Modelo de datos

El modelo de datos resultante en la tercera fase será el siguiente:

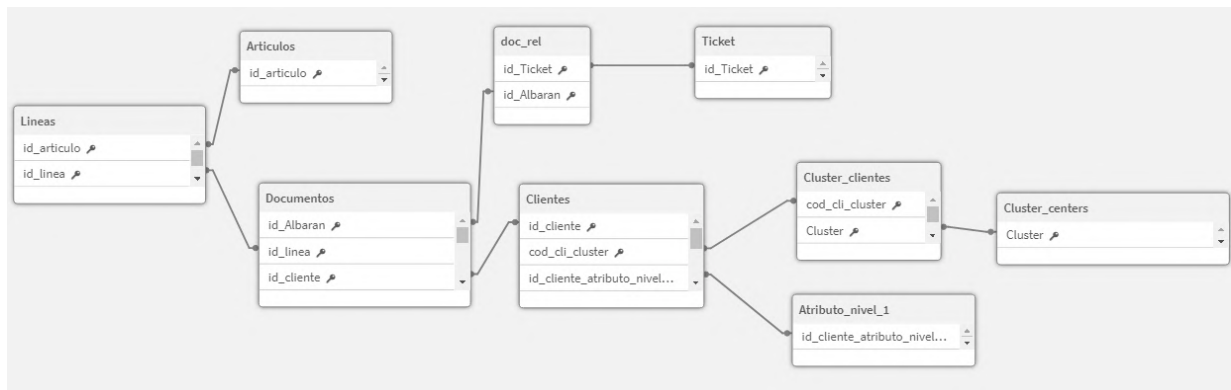


Ilustración 5: Modelo de datos tercera fase

En el modelo de datos superior podemos apreciar diferencias claras con el modelo de datos de la primera fase, ya que la forma de trabajar con las tablas cargadas en la base de datos será diferente. Ahora tenemos que cargar los artículos para todos los documentos nombrados anteriormente, por lo que deberemos estructurarlo de otra forma.

La idea del diseño de este modelo de datos ha sido juntar las cabeceras de los presupuestos, pedidos y albaranes en la tabla Documentos y las líneas de estos mismos documentos en la tabla Líneas. De esta forma, podremos relacionar todas estas líneas con una única tabla de Artículos y así, evitamos cargar tres veces la misma tabla de artículos, una para cada documento.

4.3 Creación del grado de satisfacción

Para desarrollar las tres medidas que conjuntamente nos darán el grado de satisfacción de cada cliente y serán la entrada del algoritmo de clustering de la segunda fase, se deberán tener en cuenta los tres últimos años de los presupuestos y albaranes, y los dos últimos años de los tickets generados.

Para ello, antes de crear estas medidas que representarán la evolución de los tres documentos nombrados anteriormente, se debe conocer la información anual de los documentos por separado. Con esto quiero decir que, para ejecutar la herramienta este año 2021, debemos conocer el porcentaje de presupuestos aceptados en los años 2018, 2019 y 2020 por separado, el importe de los albaranes en los años 2018, 2019 y 2020 por separado, y el número de incidencias documentadas en el año 2019 y 2020 por separado.

De esta forma, podremos observar si está aumentando o disminuyendo el contenido en estos documentos por cada cliente para, finalmente, poder categorizar el nivel de evolución de los documentos a lo largo de estos años.

Como se ha comentado al inicio del apartado, se crearán tres medidas que representarán este nivel de desarrollo, y su valor se asignará a partir de los resultados obtenidos en los años anteriores. El nivel de éxito de los presupuestos y el nivel del importe de los albaranes se calcularán de una forma similar, ya que los dos documentos tienen disponibles sus tres últimos años.

En la siguiente ilustración podremos ver como se calculará el valor de nivel para estos dos documentos:

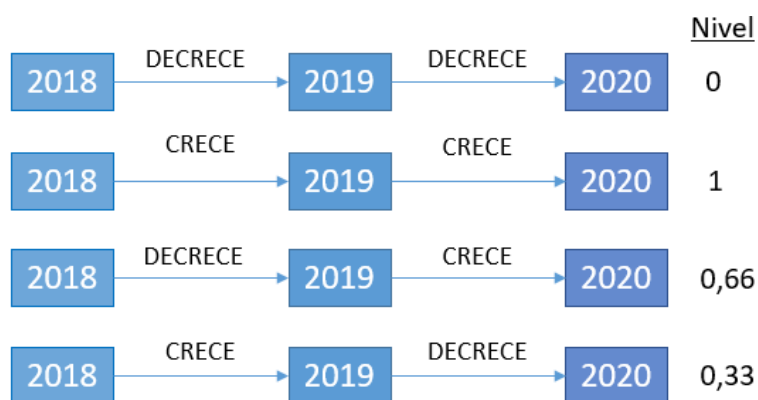


Ilustración 6: Nivel de desarrollo de los tres últimos años

Cogiendo como ejemplo el Nivel de importe de los albaranes, podemos observar que aplicamos los valores dependiendo del desarrollo que ha tenido el cliente a lo largo del tiempo. Si el cliente ha aumentado su gasto en SEMIC en cada año, entonces la categorizamos el campo de Nivel de importe para ese cliente con el valor 1, que representará el máximo valor. Por otro lado, si el cliente ha ido disminuyendo su inversión en SEMIC consecutivamente durante el periodo de tres años, lo categorizamos como valor 0, que será el valor mínimo.

Por otro lado, tenemos el nivel de evolución en los tickets, en los que no tendremos documentados correctamente los tres últimos años y este será el principal motivo por el que seleccionamos solo los dos últimos años de estos documentos. Cuando llegue el momento de ejecutar la herramienta en el año 2022 podremos actualizar esta parte, añadiendo los tres últimos años de los tickets, pero a día de hoy aun no es posible.

Entonces, la asignación de valores para la evolución de los dos últimos años de los tickets será la siguiente:

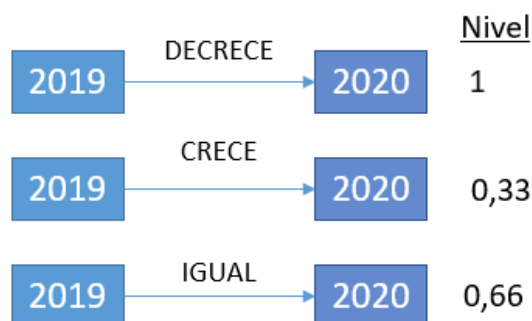


Ilustración 7: Nivel de desarrollo de los Tickets

Como podemos observar en la ilustración anterior, la principal diferencia en cuanto a los otros dos documentos, aparte de que solo encontramos los dos últimos años, será que si las incidencias crecen a lo largo del tiempo será negativo, mientras que si decrecen será positivo. Entonces, si las incidencias crecen de 2019 a 2020 serán categorizadas con el valor 0.33, si decrecen con el valor 1 y si son iguales con el valor 0.66.

4.4 Ejecución de la herramienta

La herramienta diseñada en este proyecto está pensada para que sea ejecutada una vez al año o si ha habido un cambio significativo en los clientes que se encuentran en la base de datos de la empresa, aunque esto segundo no será habitual.

Como la herramienta utiliza los datos de los tres años anteriores, se podría ejecutar durante la primera semana de enero para poder conocer, cuanto antes, si se ha mejorado o no en la satisfacción de los clientes y conocer las principales características y causas de la situación de estos.

5 Implementación

En este apartado se va a tratar sobre el software y las herramientas utilizadas para desarrollar todas las fases el proyecto.

5.1 Qlik Sense

Qlik Sense es una plataforma que engloba todos los procesos necesarios para la creación de una aplicación de *Business Intelligence* para el análisis de datos. Permite al usuario combinar diferentes fuentes de datos en una única vista y poder consultar la información de forma sencilla sin necesidad de realizar nuevas consultas a la base de datos.

La plataforma de Qlik Sense se divide en dos partes principales, por un lado tenemos un editor de carga de datos donde podremos realizar la fase de ETL (*Extract, Transform, Load*) en su totalidad. Por otro lado, contamos con una plataforma de visualización de los datos, que nos permite visualizar la información cargada anteriormente de forma muy sencilla, mediante la funcionalidad de arrastrar y soltar.

También, permite al usuario filtrar la información directamente desde la visualización y es capaz de lograr una interactividad en tiempo real sin que se tenga que realizar ninguna consulta extra. Cada vez que el usuario realiza una selección en los datos, estos se deben recalcular para que respeten la selección. La forma de realizar este recalcule de datos la definirá el modelo de datos, en el que se definen las relaciones entre las diferentes tablas y, por lo tanto, se tratará de un modelo relacional.

5.1.1 Script de carga de datos

Qlik Sense utiliza un script de carga de datos para poder conectarse a diversas fuentes de datos y recuperar los datos alojados en ellas. En él, se especifican los campos y las tablas que se van a cargar, y se puede manipular la estructura de datos mediante sentencias.

Durante la carga de datos, Qlik Sense identifica los campos comunes en distintas tablas para asociar los datos, por lo que se deberá tener claro que campos de relación entre tablas van a existir para nombrarlos de la misma forma.

El análisis en Qlik Sense siempre se produce mientras la app no está directamente conectada con sus fuentes de datos, estos se almacenan en memoria en la base de datos interna de Qlik Sense.

5.1.1.1 Sintaxis de script

Como se ha comentado, un script se compone de una serie de sentencias y palabras clave que se ejecutan de forma consecutiva. En él, se define el nombre de la fuente de datos, los nombres de las tablas y los nombres de los campos que queremos cargar.

A continuación veremos las funciones y expresiones más utilizadas en el script del proyecto: [2]

- Uso de la sentencia Qualify y Unqualify:

La sentencia Qualify se utiliza para que los nombres de los campos obtengan el nombre de la tabla como un prefijo, de esta forma la unión automática entre campos con el mismo nombre en diferentes tablas se puede suspender. Para evitar que se creen relaciones no deseadas esta sentencia es vital.

El nuevo nombre tendrá la forma: *nombretabla.nombrecampo*, donde nombretabla es equivalente a la etiqueta de la tabla actual, o, si no hay ninguna etiqueta, al nombre que aparece después de From en sentencias LOAD y SELECT.

La calificación que se crea con Qualify, se puede desactivar mediante la sentencia Unqualify.

Estas dos sentencias tienen un argumento que es el listado de los campos separados por comas en los que se tiene que aplicar la calificación. También está permitido el uso del símbolo * para indicar la totalidad de los campos.

- Mapping Load y ApplyMap:

El prefijo Mapping ofrece una manera eficaz de sustituir valores de campo durante la ejecución del script.

Se puede colocar delante de una sentencia LOAD o SELECT y almacenará el resultado de la sentencia de carga como una tabla de enlace. Esta se compone por dos columnas, la primera contiene los valores de comparación y la segunda contiene los valores de correspondencia deseados.

Por ejemplo: se podría utilizar para reemplazar los valores de US, U.S. o América por USA.

Las tablas de correspondencia se almacenan temporalmente en la memoria y se eliminan automáticamente tras la ejecución de script.

La función de script ApplyMap sirve para enlazar (o mapear) el resultado de una expresión con una tabla de correspondencia previamente cargada.

Su sintaxis es:

ApplyMap ('map_name', expression [, default_mapping])

- *Map_name*: Es el nombre de una tabla de asignación que se creó previamente a través de la sentencia mapping load o mapping select. El nombre se le da antes de realizar la sentencia y termina con dos puntos (:).
- *Expression*: Es la expresión que se debe mapear o asignar.
- *Default_mapping*: Si se establece, este valor se usará como valor predeterminado si la tabla de asignación no contiene un valor coincidente para expresión.

- Función condicional IF:

La función if devuelve un valor dependiendo de si la condición proporcionada con la función se evalúa como True o False. Su sintaxis es la siguiente:

If(condition , then , else)

5.1.2 Requisitos del sistema Qlik Sense

En este apartado se nombrarán los requisitos que debe cumplir el sistema para instalar y ejecutar correctamente Qlik Sense, en concreto para la versión de Febrero de 2021. [2]

5.1.2.1 Plataformas

- Microsoft Windows Server 2012 R2
- Microsoft Windows Server 2016
- Microsoft Windows Server 2019

5.1.2.2 Procesadores

Los procesadores requeridos deberán ser multinúcleo compatibles con x64. El fabricante recomienda utilizar al menos 4 núcleos por nodo.

5.1.2.3 Memoria

La capacidad de la memoria deberá ser de al menos 8 GB, pero dependiendo del volumen de los datos podría requerirse más. Como Qlik Sense es una tecnología de análisis en memoria, los requisitos de esta van directamente relacionados con la cantidad de datos que se esté analizando.

5.1.2.4 Espacio en el disco

El total requerido para realizar la instalación es de 5 GB.

5.1.2.5 Navegadores compatibles de Microsoft Windows

Los siguientes navegadores se pueden utilizar en máquinas Microsoft Windows y Microsoft Windows Server y serán compatibles para acceder a Qlik Management Console (QMC), que se utiliza para administrar Qlik Sense, y al centro de control de Qlik Sense.

- Microsoft Edge (solo para Microsoft Windows 10)

- Google Chrome
- Mozilla Firefox

5.1.3 Características del sistema del proyecto

En el proyecto se ha utilizado Qlik Sense sobre el siguiente hardware y software:

Un servidor con CPU AMD EPYC 7302P 16-Core, con una capacidad de memoria RAM de 24 GB. El servidor funciona sobre el sistema operativo Windows Server 2012 R2 de 64 bits. Para acceder al centro de control de Qlik Sense se ha utilizado el navegador Google Chrome.

5.1.4 Conexión a las fuentes de datos

Qlik Sense dispone de múltiples posibilidades para conectarse a diferentes fuentes de datos, algunos ejemplos podrían ser MongoDB, MySQL, PostgreSQL, y un largo etcétera. Por otro lado, también permite acceder a ubicaciones de archivos de nuestro equipo o de Dropbox.

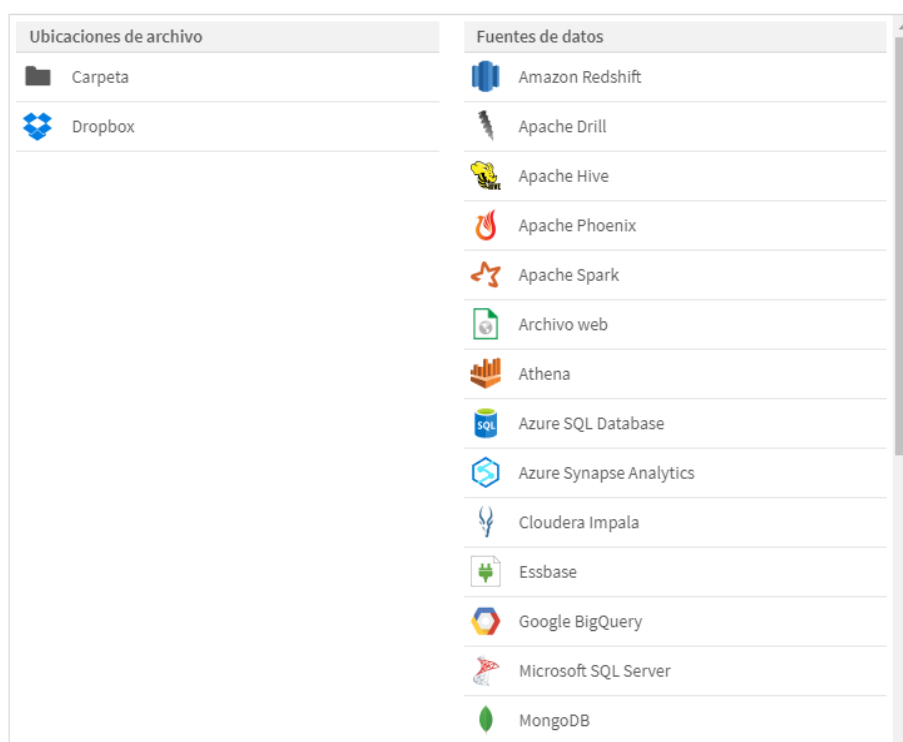


Ilustración 8: Algunas fuentes de datos para crear una conexión en Qlik Sense

Para desarrollar este proyecto se utilizará el conector ODBC (*Open Database Connectivity*) para realizar la conexión a la base de datos.

ODBC es una API estandarizada que permite la conexión a los servidores de base de datos SQL y, por lo tanto, utiliza el lenguaje estructurado de consultas SQL como lenguaje de acceso a la base de datos. [3]

Los paquetes de ODBC están ya integrados en Qlik Sense, lo que hace que resulte muy sencillo establecer la conexión mediante esta herramienta. Además, Qlik Sense proporciona un buscador para seleccionar los datos, en el que se pueden buscar las tablas y nos muestra una vista previa de los campos que contiene y del SQL resultante para recuperarlos.

También, se deberá establecer la conexión con la ubicación del servidor que contenga los archivos Excel resultantes de la fase 2 para que puedan ser cargados y relacionados con ciertas tablas de la base de datos en Qlik Sense.

5.1.4.1 Base de datos utilizada en el proyecto

La plataforma de gestión de datos a la que nos conectaremos desde Qlik Sense para poder realizar las consultas y recuperar la información necesaria para desarrollar el proyecto será Informix.

Informix, es un gestor de base de datos que incluye un sistema de administración de base de datos relacionales basado en SQL y está desarrollado por IBM.

5.1.5 Secciones del script y transformación de los datos

- Main:

En esta sección se recogen las configuraciones generales del script, como por ejemplo los formatos por defecto para horas, días de la semana, meses, moneda, etc. Estas configuraciones se generan automáticamente al crear la aplicación de Qlik Sense.

También, se encuentran la declaración de la conexión ODBC a la base de datos y la creación de variables globales que se utilizarán para filtrar los datos al cargarlos mediante consultas a la base de datos.

En el proyecto utilizaremos dos variables globales:

- *vInicio*: Esta variable será el año a partir del cual vamos a cargar los datos. Como necesitamos obtener los presupuestos, albaranes y tickets de los últimos tres años completos, el valor de esta variable será el año en el que se ejecuta la herramienta menos 3, de esta forma se recuperarán siempre los datos correctos independientemente del año en el que se ejecuten. Esta variable la utilizaremos en los SQLs donde carguemos alguno de los tres documentos nombrados anteriormente.
- *vFinal*: Esta variable será el año límite del cual vamos a cargar los datos. Y se utilizará en los SQLs al igual que la variable de *vInicio*. Esta variable tendrá el valor del año de ejecución de la herramienta menos 1.
- *Empresas*: Esta variable será el número de la empresa de la que se van a cargar los datos. Como se ha visto anteriormente, la empresa de la que vamos a realizar el análisis es SEMIC y su número de empresa dentro del ERP es el 1, por lo tanto, esta variable será igual a 1. Se utilizará en todos los SQLs, ya que cada empresa tiene sus propios documentos y sus propios clientes. Si algún día se quiere realizar un análisis de otra empresa del grupo simplemente se deberá modificar el valor a esta variable y automáticamente las consultas a la base de datos serán de la empresa deseada.

- Autocalendar:

En esta sección, se generan los diferentes formatos que se pueden aplicar a las fechas, esto se realiza gracias a *autoCalendar* y se puede añadir a todos los campos que sean fechas del script.

Su función será generar una extensión de los campos de fecha que podremos utilizar en las visualizaciones. Por ejemplo, se tendrá la opción de recuperar solo el año o mes de la fecha en vez de la fecha al completo.

Las fechas que se han declarado dentro de *autoCalendar* para que se genere las extensiones con los diferentes formatos, son las fechas de la cabecera del presupuesto, la de la cabecera del albarán y la fecha de alta del ticket. Se utilizará la extensión que nos devuelve solamente el año de la fecha para agrupar los datos por años en la visualización.

- Mappings:

En esta sección se encontrarán todos los Mapping realizados. Para que estos funcionen correctamente, deben crearse antes de utilizarlos en el ApplyMap, por este motivo los creamos en una sección que se ejecuta al inicio de la carga de datos.

Para recuperar la información de, si a través de un presupuesto se ha llevado a cabo un pedido necesitamos una tabla de mapping, ya que queremos recuperar información de otras tablas sobre la tabla de presupuestos.

Para poder obtener esta información, se conoce que desde el presupuesto no se tiene ningún dato del pedido, pero desde el pedido sí que se tiene documentado el presupuesto del que proviene.

Por ello, si se cargan todos los presupuestos que están documentados en la tabla de cabeceras del pedido, mediante el identificador del presupuesto desde la tabla de cabeceras de presupuestos, se podrá obtener un campo que nos de la información de si ese presupuesto tiene pedido o no, y de esta forma poder utilizarlo para calcular la medida de éxito de presupuestos.

```

Pedidos_presupuesto:
Mapping Load
    empresa & '-' & ser_pre & '-' & n_preven,
    'Tiene pedido';
SQL SELECT empresa, ser_pre, n_preven
FROM usadm.pedcab
WHERE Year(fecha)>= $(vInicio);

```

Ilustración 9: Tabla de mapeo de presupuestos con pedido

Como se puede observar en la imagen anterior, se carga la tabla de cabeceras de pedidos (*pedcab*), con los pedidos realizados desde 2018 (variable *vInicio* en el año de ejecución 2021) hasta la actualidad.

Esta tabla tiene documentado el presupuesto en los campos de *ser_pre* y *n_preven*. Por lo tanto, si el identificador de presupuesto que le pasamos en el ApplyMap coincide con alguno de los que ha cargado la tabla de Mapping, entonces devolverá ‘Tiene pedido’, sino no devolverá nada.

En este momento, puede surgir la duda de porque no se acota también la fecha máxima del pedido con la variable de *vFechaFinal*, creada en la sesión Main. El motivo es que puede ser que un presupuesto de 2020 haya generado un pedido en lo que llevamos de 2021, por lo tanto, será un presupuesto exitoso. Si hubiéramos acotado con la fecha máxima aparecería como un presupuesto “fallido”.

- Cliente Empresa Asociada:

En esta sección, se creará manualmente la tabla con los códigos de cliente que tienen las empresas que pertenecen al grupo, ya que no se tienen identificadas en la base de datos. Tras esto, se creará una tabla Mapping con la que poder identificar si un cliente es una empresa asociada al grupo o no.

En Qlik Sense tenemos la posibilidad de crear tablas manualmente mediante la sentencia *load inline*. Como se conoce con anterioridad el código de cliente con el que están documentadas estas empresas, simplemente se deben introducir estos códigos en la tabla junto con el código de empresa de SEMIC.

La tabla de mapeo se alimentara de los datos introducidos manualmente en la tabla anterior y nos será útil para marcar los documentos que pertenezcan a las empresas del grupo.

Esto lo realizaremos gracias al siguiente código:

```
Clientes_grupo:
load * inline [
empresa, cod_cliente
1, "63328"
1, "0327"
1, "48790"
1, "75834"
1, "78983"
];

Cliente_empresa_asociada:
Mapping LOAD empresa &'-'& cod_cliente,
'EMPRESA ASOCIADA' AS empresa_asociada Resident Clientes_grupo;

Drop Table Clientes_grupo;
```

Ilustración 10: Creación y mapeo de la tabla con los clientes del grupo

Como se puede ver en el fragmento del script, el campo de empresa siempre será 1, ya que se realiza el análisis de la empresa SEMIC. Por otro lado, el campo código cliente serán los códigos de los clientes que pertenecen a SEMIC, pero que son empresas del grupo.

A la tabla de mapeo, se le pasará el campo de empresa y el código de cliente que se está procesando en ese momento. Si ese código de cliente aparece documentado en la tabla que se ha creado manualmente, entonces devolverá 'EMPRESA ASOCIADA', sino no devolverá nada. De esta forma, se podrá tener un campo de referencia, el cual utilizaremos para excluir de las visualizaciones los datos que pertenezcan a estas empresas.

- Clientes:

En esta sección, cargaremos la tabla de clientes para poder obtener el código de los clientes de SEMIC, y la tabla de verticales, para poder filtrar los clientes por la vertical de Indeterminado.

Como ya se ha comentado, en el caso de los clientes, se necesita recuperar los campos de empresa, código de cliente, y N.I.F. del cliente. También se requiere que los clientes pertenezcan a la empresa 1, y que no tengan fecha de baja o, si la tienen documentada, que esta sea mayor a la fecha final del análisis. De esta forma, se puede asegurar que los clientes no estarán dados de baja durante el periodo sobre el que realizamos el análisis.

Tras tener claro que clientes se deben cargar, es el momento de crear todas las claves para relacionar la tabla de clientes con el resto de tablas. La clave primaria de clientes se formará mediante el campo *empresa* y el campo *cod_cli*, el cual será el código del cliente. La clave primaria tendrá el mismo formato para todas las relaciones con el resto de tablas, pero se debe modificar el nombre para cada relación, ya que Qlik Sense genera las relaciones entre tablas basándose en el nombre del campo.

Así que, las relaciones iniciales serán las siguientes:

```
/* Relaciones con las demas tablas */
empresa &'-'& cod_cli as id_cliente_presupuestos,
empresa &'-'& cod_cli as id_cliente_facturacion,
empresa &'-'& cod_cli as id_cliente_atributo_nivel_1,

cod_cli as codigo_cliente,
```

Ilustración 11: Campos de relación de la tabla de Clientes

Donde el campo *id_cliente_presupuestos* será la relación con la tabla de cabecera de presupuestos, el campo *id_cliente_facturación* será la relación con la tabla de cabecera de albaranes y finalmente, el campo *id_cliente_atributo_nivel_1* será la relación con la tabla de atributos del cliente.

Por otro lado, se debe crear un campo que sirva para identificar si un cliente es una persona física o una empresa. Se utilizará el N.I.F del cliente el cual estará documentado en el campo *nif_cli*. Como se ha comentado, el documento de identificación de una persona física puede empezar o por un número, o por las letras X o Y, por lo que si cumple una de estas tres condiciones, el cliente será una persona física y sino, será una empresa.

Para llevarlo a cabo se utilizará la función *Left (nombre_del_campo, 1)* que devolverá el primer carácter empezando por la izquierda del campo que se le pasa en el primer parámetro, es decir, del campo *nif_cli*.

```
/* Campo para filtrar en la visualización a los clientes que sean personas físicas de los que no */  
If(IsNum(Left(nif_cli, 1)) or Left(Upper(nif_cli), 1) = 'X' or  
   Left(Upper(nif_cli), 1) = 'Y', 'ES PERSONA FISICA', 'ES EMPRESA') as es_persona_fisica
```

Ilustración 12: Tratamiento para identificar si el cliente es una persona física

En cuanto a la tabla de atributos de nivel 1 del cliente, también denominados en SEMIC como vertical. Se debe crear una tabla a parte de la de clientes, y relacionarlas entre sí mediante los campos empresa y código de cliente, en este caso el campo de relación se denominará *id_cliente_atributo_nivel_1*.

El motivo de crear una nueva tabla y no una tabla de mapeo, es que un cliente puede tener más de un atributo de nivel 1 y si se utilizara una tabla de mapeo, esta devolvería solamente uno de los dos. Como en el proyecto se requiere la vertical de Indeterminado no debería haber ningún problema, ya que supuestamente no es compatible con otras verticales, pero de esta forma se asegura un correcto funcionamiento.

- Presupuestos:

En esta sección, se carga la tabla donde se encuentran los datos de las cabeceras de los presupuestos. Se deben cargar todos los presupuestos que estén entre el rango de fechas y que pertenezcan a la empresa 1. Los campos que se necesitan recuperar son el de empresa, código de cliente, la serie y el número que identificará al presupuesto y la fecha del presupuesto.

Se deberá crear la clave de relación con la tabla de clientes que se denominará *id_cliente_presupuestos*. También, se recupera la fecha para poder filtrar por los datos de un año u otro, la serie del presupuesto, la cual se utilizará como una parte de la clave para averiguar si tiene pedido, y el número de presupuestos, que utilizaremos para poder distinguir el número de presupuestos que tenemos por

cliente y con esto, generar la medida de éxito de presupuestos que será presupuestos exitosos dividido de los presupuestos totales.

Además, es necesario conocer si los clientes de los presupuestos son empresas del grupo o no, para esto se utilizará la tabla de mapeo creada en la sección de Cliente Empresa Asociada. De esta forma, el campo que se denominará *factu_consolidada* será de utilidad para excluir a las empresas del grupo.

```
ApplyMap('Cliente_empresa_asociada', empresa&'-'&cod_cli, 'CLIENTE EXTERNO') as factu_consolidada,
```

Ilustración 13: ApplyMap cliente es una empresa del grupo

Para finalizar, se debe conocer si un presupuesto tiene pedido o no. Para ello, se utiliza la tabla de mapeo *Pedidos_presupuesto*, creada en la sección de Mappings. A esta, se le pasa el identificador del presupuesto que constará de empresa, serie y número, todas estas concatenadas con un guion, y si ese presupuesto tiene algún pedido relacionado devolverá *‘Tiene pedido’*, sino devolverá *‘Presupuesto sin pedido’*.

```
ApplyMap('Pedidos_presupuesto', empresa&'-'&cod_ser&'-'&n_preven, 'Presupuesto sin pedido') as Tiene_pedido,
```

Ilustración 14: ApplyMap presupuesto tiene pedido

- **Albaranes:**

En esta sección se carga la tabla de las cabeceras de los albaranes. Los campos que se deben recuperar son *empresa*, código del cliente y *fecha* que será la fecha de creación del albarán. También, *cod_ser* y *n_albven* que serán la serie y el número del albarán.

En este caso, las cabeceras de los albaranes deberán ir relacionadas con las líneas del albarán, las cuales se necesitan para obtener el importe del albarán, la relación será denominada como *id_alblin*. Además, se relacionará con la tabla de Clientes mediante el campo *id_cliente_facturacion*, como ya se había comentado en la sección de Clientes. Y finalmente, se relacionará con la tabla intermedia que permitirá

acceder a los tickets generados a partir de la venta al cliente, mediante el campo *id_Albaran*.

Finalmente, se creará el campo de *factu consolidada* de la misma forma que en la sección de presupuestos.

- Líneas:

En esta sección se encuentran las líneas de los albaranes, de la cuales se recuperan los campos de *empresa*, *cod_ser*, *n_albven* que serán la relación con las cabeceras de albaranes y también el importe de cada línea del albarán.

```
SQL SELECT T0.empresa, T0.cod_ser, T0.n_albven, T0.importe, T0.cod_iva, T0.imp_adi
FROM usadm.alblin T0
INNER JOIN usadm.albcab T1
ON T1.empresa = T0.empresa AND T1.cod_ser = T0.cod_ser AND T1.n_albven = T0.n_albven
AND T1.empresa in $(empresas) AND year(T1.fecha)>=$(vInicio) AND year(T1.fecha)<=$(vFinal)
WHERE T0.cod_art IS NOT NULL AND T0.unidades <> 0 AND T0.empresa IN $(empresas);
```

Ilustración 15: SQL de carga de las líneas del albarán

Como se observa en la imagen superior, se debe realizar una sentencia *INNER JOIN* para poder recuperar las líneas que tengan un albarán disponible en el rango de fechas definido, ya que las líneas del albarán no tienen disponible ningún campo de fecha. Estas líneas también deben cumplir que pertenezcan a la empresa 1 (SEMIC), que tengan como mínimo un artículo, es decir que el código del artículo no sea nulo, y que las unidades del artículo sean diferentes a 0.

El motivo por el que utiliza *unidades* diferentes de 0 y no *unidades* mayor que 0 es porque se pueden obtener albaranes con unidades negativas. Esto significará que se ha producido una devolución, y se devolverá el importe del albarán al cliente. Por lo tanto, se restará también del importe total que tenga el cliente en el análisis.

Tras tener claro el SQL, se debe crear la relación con las cabeceras del albarán, la cual ya sabemos que se denominará *id_alblin*. También, recuperar los campos de *imp_adi* e *importe*. Finalmente, el campo de *cod_iva* lo utilizaremos como clave para, a través de una tabla de mapeo, recuperar el tipo del I.V.A. que se le aplica a la línea.

```

Qualify *;
Unqualify id_alblin;
Lineas_Albaranes:
Left keep(Albaranes)
LOAD
    empresa &'-AV-'& cod_ser &'-'& n_albven as id_alblin,

    importe,
    imp_adi,
    applymap('Clases_iva', empresa &'-E-'& cod_iva, 'SIN IVA') as Tp_iva;

```

Ilustración 16: Tabla de Líneas de Albaranes

Como se puede observar en el fragmento de script, se utiliza también la sentencia *Left keep*. Su función será cargar solamente las líneas que estén relacionadas con la tabla de albaranes cargados anteriormente en memoria en Qlik Sense.

- Ticket:

En esta sección, se cargará la tabla de tickets que contendrá las incidencias generadas y la tabla de relación que será necesaria para poder relacionar los albaranes con sus incidencias.

La tabla de relación, la cual denominaremos *doc_rel*, se cargará desde la tabla de la base de datos *tckdoc*. Esta tiene campos donde se documenta el identificador del albarán, que serán *emp_doc*, *ser_doc* y *n_doc*, y a su vez, campos donde se documenta el identificador del ticket, que serán *empresa*, *cod_ser* y *n_ticket*.

Estos dos identificadores se utilizarán como relaciones entre tablas, el campo *id_Albarán* se relacionará con la tabla de Albaranes y el campo *id_Ticket* con la tabla de Tickets. De esta tabla no necesitamos cargar ningún campo más, ya que se trata de una tabla de paso.

```

Qualify *;
Unqualify id_Albaran, id_Ticket;
doc_rel:
LOAD emp_doc &'-'& ser_doc &'-'& n_doc as id_Albaran,
    empresa &'-'& cod_ser &'-'& n_ticket as id_Ticket;

SQL SELECT emp_doc, ser_doc, n_doc, empresa, cod_ser, n_ticket
FROM usadm.tckdoc;

Unqualify *;

```

Ilustración 17: Tabla de relación entre Albaranes y Tickets

Como se han visto varias tablas de mapeo, puede surgir la pregunta de por qué esta tabla intermedia no se crea como una tabla de mapping. La respuesta es que una tabla de mapping te devuelve un registro por cada consulta que se le realiza. En este caso, un albarán puede tener más de un ticket relacionado y si se carga como una tabla de mapeo devolvería solo un ticket de los relacionados.

En cuanto a la tabla de Tickets, se necesita recuperar el identificador del ticket, es decir, los campos de *empresa*, *cod_ser* y *n_ticket*, y el campo *fecha_a*, que representará la fecha en la que se generó el ticket.

Las condiciones que un ticket debe cumplir para ser cargado son, que pertenezca a la empresa 1, que la fecha de alta sea mayor a la de inicio y finalmente que pertenezcan a los departamentos de logística o administración, esto se seleccionará mediante los códigos que representan a cada departamento a través del campo *cod_dpe*. El motivo de seleccionar solamente estos dos departamentos es que son los únicos departamentos que empezaron a documentar tickets a partir de 2019.

En la siguiente imagen se puede apreciar la sentencia SQL:

```
SQL SELECT empresa, cod_ser, n_ticket, fecha_a
FROM usadm.ticket
WHERE empresa IN $(empresas) AND year(fecha_a) >= $(vInicio) AND cod_dpe IN ("0001","0005");
```

Ilustración 18: SQL de carga de los tickets

Para terminar, se debe crear la relación con la tabla intermedia mediante el campo *id_Ticket*. También, se debe recuperar el campo de *n_ticket*, que se utilizará para contar el número de tickets que se tiene por cliente, ya que este campo será único para cada ticket.

Podremos ver el script de carga en Qlik Sense en la siguiente imagen:

```
Qualify *;
Unqualify id_Ticket;
Ticket:
LOAD
    empresa & '-' & cod_ser & '-' & n_ticket as id_Ticket,
    n_ticket,
    fecha_a;
```

Ilustración 19: Tabla de Tickets

5.1.6 Creación de las medidas

Como se ha comentado, se necesita separar el porcentaje de presupuestos aceptados de los tres últimos años, el importe de los albaranes de los tres últimos años y los tickets generados en los últimos dos años. Así que, se creará una columna de la tabla para cada año de cada documento y de esta forma, podremos comparar entre ellos para ver si su valor crece o decrece. También, se deberá tener una columna con el código del cliente para poder visualizar el valor de las medidas agrupadas por cada cliente.

Para poder separar los datos correspondientes en las diferentes columnas, las medidas se crean desde una expresión compuesta de funciones de agregación, como por ejemplo Sum o Count. Estas, también tienen la posibilidad de aplicar ciertos filtros internos, lo que en Qlik Sense se denomina como *Set Expression*. Se utilizarán para filtrar a los presupuestos que tengan uno o más pedidos relacionados, a los clientes que no sean empresas del grupo y el año del que se muestran los datos.

Todos los filtros de campo que se quieran aplicar irán entre los símbolos {<...>}, y para seleccionar el valor concreto que debe tener ese campo se introduce entre corchetes y comillas simples.

En el siguiente ejemplo, se podrá observar la *Set Expression* que nos devolverá el porcentaje de éxito de los presupuestos en el año 2018.

```
Count({<[Presupuestos.Tiene_pedido] = {'Tiene pedido'}, [Presupuestos.fecha.autoCalendar.Year] = {'2018'},  
[Presupuestos.factu_consolidada] = {'CLIENTE EXTERNO'}>} distinct [Presupuestos.n_preven])  
/  
Count({<[Presupuestos.fecha.autoCalendar.Year] = {'2018'}, [Presupuestos.factu_consolidada] = {'CLIENTE EXTERNO'}>}  
distinct [Presupuestos.n_preven])
```

Ilustración 20: Ejemplo de Set Expression de Éxito Presupuestos 2018

En la *Set Expression* anterior, se puede observar que se realiza la división entre los presupuestos exitosos y los presupuestos totales. Para ambas partes, se seleccionan los presupuestos con la fecha en el año 2018 y los clientes externos, es decir, los que no pertenecen al grupo, gracias al campo que se ha creado de *factu_consolidada*.

En la parte superior de la división, se filtran los presupuestos que tienen pedido, esto sucederá cuando el campo *Tiene_pedido* tenga el valor de 'Tiene pedido'.

Para contar el número de presupuestos distintos se utilizan las funciones *Count* y *distinct*, aplicadas al número identificador del presupuesto, denominado *n_preven*.

Este tratamiento, se debe realizar para los tres años que se tienen en cuenta para el análisis. De esta forma, tendremos los valores separados en tres columnas y podremos comparar entre ellos para asignar un valor al nivel de éxito de los presupuestos, dependiendo si crecen o decrecen sus valores a lo largo de estos años.

Quedando la tabla de la siguiente forma:

Código Cliente	Q	Exito Presupuestos 2018	Exito Presupuestos 2019	Exito Presupuestos 2020
Totales				
0001	-	-	-	0,00
0004		0,55	0,42	0,63
0008	-	-	-	-
0020	-	-	-	-
0026		0,50	0,00	0,00
0044		0,80	-	0,00
00049		0,00	-	-
0050		0,50	0,33	-
0051		0,00	0,67	0,25
0055	-	-	0,00	0,30

Ilustración 21: Medidas de éxito de presupuestos

Como se puede observar, en algunos clientes el cálculo devuelve valores nulos que se deben tratar. Se ha decidido tratarlo en la visualización de Qlik Sense, y se llevará a cabo mediante la siguiente operación condicional: Si el resultado de la división mostrada anteriormente es nulo, entonces se devolverá un 0 para que se muestre en el campo con valor nulo, sino mostraremos el resultado de la división. Por lo tanto, el ejemplo mostrado anteriormente de la columna *Éxito Presupuestos 2018* quedaría de la siguiente forma:

```

If(IsNull(Count({<[Presupuestos.Tiene_pedido] = {'Tiene pedido'}, [Presupuestos.fecha.autoCalendar.Year] = {'2018'},
[Presupuestos.factu_consolidada] = {'CLIENTE EXTERNO'}} distinct [Presupuestos.n_preven]))
/
Count({<[Presupuestos.fecha.autoCalendar.Year] = {'2018'}, [Presupuestos.factu_consolidada] = {'CLIENTE EXTERNO'}}
distinct[Presupuestos.n_preven])), 0,

Count({<[Presupuestos.Tiene_pedido] = {'Tiene pedido'}, [Presupuestos.fecha.autoCalendar.Year] = {'2018'},
[Presupuestos.factu_consolidada] = {'CLIENTE EXTERNO'}} distinct [Presupuestos.n_preven]))
/
Count({<[Presupuestos.fecha.autoCalendar.Year] = {'2018'}, [Presupuestos.factu_consolidada] = {'CLIENTE EXTERNO'}}
distinct [Presupuestos.n_preven]))

```

Ilustración 22: Ejemplo de Set Expression de Éxito Presupuestos 2018 con valores nulos tratados

Como se puede observar, en la primera parte se comprueba si la división es nula mediante *IsNull*, si es nula se devolverá el valor 0 y, si no se volverá a realizar la división para devolver su valor.

Tras realizar este tratamiento, los campos de éxito de presupuestos en la tabla quedarán de la siguiente forma:

Código Cliente	Exito Presupuestos 2018	Exito Presupuestos 2019	Exito Presupuestos 2020
Totales			
0001	0,00	0,00	0,00
0004	0,55	0,42	0,63
0008	0,00	0,00	0,00
0020	0,00	0,00	0,00
0026	0,50	0,00	0,00
0044	0,80	0,00	0,00
0049	0,00	0,00	0,00
0050	0,50	0,33	0,00
0051	0,00	0,67	0,25
0055	0,00	0,00	0,30

Ilustración 23: Medidas de éxito de presupuestos con valores nulos tratados

En cuanto a las columnas de importe de albaranes, se crearán mediante la siguiente *Set Expression*:

```
Sum( {<[Albaranes.fecha.autoCalendar.Year] = {'2018'}, [Albaranes.factu_consolidada] = {'CLIENTE EXTERNO'}}
If(Albaranes.Sem_iva = 'N',
    Lineas_Albaranes.importe,
    (Lineas_Albaranes.importe / (1 + (Lineas_Albaranes.Tp_iva/ 100)))))
```

Ilustración 24: Set Expression de Importe Albaranes 2018

Se puede observar que se trata de la columna que hace referencia a los albaranes del año 2018 y que también se utiliza el filtro de que el cliente sea un cliente externo al grupo.

Para tratar el importe de los albaranes correctamente se debe obtener el importe del artículo sin el I.V.A., ya que puede ser que a diferentes productos o clientes se le apliquen cantidades diferentes de I.V.A y entonces no serían importes reales para la empresa.

Por lo tanto, la realización del cálculo para obtener el importe correctamente, se lleva a cabo de la siguiente forma. Si el semáforo que nos indica si el I.V.A. esta

aplicado en el importe o no (*Sem_iva*) es igual a N, querrá decir que no tiene el I.V.A. aplicado y por lo tanto, se devolverá el valor del campo importe directamente.

Por otro lado, si el semáforo no es N, querrá decir que sí que tiene el I.V.A. aplicado y se deberá deshacer el cálculo de este, ya que se quiere recuperar importes sin I.V.A. En el campo *Tp_iva* encontramos la cantidad de I.V.A. que se le ha aplicado, por lo tanto, si dividimos esa cantidad entre 100 y le sumamos 1 obtenemos la cantidad por la que se dividirá el importe para obtener el importe del producto sin I.V.A.

Finalmente, las columnas que muestran los datos anuales de los tickets son las más sencillas de generar, ya que en la *Set Expression* simplemente se debe filtrar por el año de la fecha de alta de la incidencia y contar el numero diferente de tickets mediante las funciones *Count* y *distinct* aplicadas al número que identifica al ticket, el cual encontramos en el campo *n_ticket*.

Se puede apreciar el cálculo de la medida de Tickets 2019 en la siguiente imagen:

```
Count(<{<[Ticket.fecha_a.autoCalendar.Year] = {'2019'}>> distinct [Ticket.n_ticket])
```

Ilustración 25: Set Expression de Tickets en 2019

Recordar que en los ejemplos anteriores se han visto las medidas para un año en concreto, el resto de años se calcularán de forma similar a su ejemplo, simplemente se debe realizar el cambio del año que se vaya a calcular dentro de la *Set Expression* de cada columna.

Al finalizar las medidas de cada año con sus respectivos datos, se obtendrá la siguiente visualización:

Código Cliente	Exito Presupuestos 2018	Exito Presupuestos 2019	Exito Presupuestos 2020	Importe Albaranes 2018	Importe Albaranes 2019	Importe Albaranes 2020	Ticket LO y AD 2019	Ticket LO y AD 2020
Totales								
0001	0,00	0,00	0,00	0,00	0,00	0,00	0	0
0004	0,55	0,42	0,63	11.946,31	10.765,80	11.272,84	3	2
0008	0,00	0,00	0,00	0,00	0,00	49,46	0	0
0020	0,00	0,00	0,00	0,00	0,00	0,00	0	0
0026	0,50	0,00	0,00	2.706,29	1.619,70	0,00	0	0
0044	0,80	0,00	0,00	5.776,08	1.494,95	1.852,95	0	3
00049	0,00	0,00	0,00	0,00	0,00	0,00	0	0
0050	0,50	0,33	0,00	1.177,48	4.656,59	3.487,94	1	7
0051	0,00	0,67	0,25	2.801,46	3.367,58	2.707,26	0	1
0055	0,00	0,00	0,30	0,00	0,00	2.755,56	0	0

Ilustración 26: Tabla de datos con medidas anuales

Tras analizar los valores obtenidos en la tabla resultante, se ha observado que hay bastantes clientes que tienen todos sus campos a cero, esto podrían ser clientes antiguos que en los tres años de los que recuperamos datos no han realizado ninguna actividad con SEMIC. Sería de utilidad crear un campo que ayude a identificar a estos clientes inactivos, ya que no se deben procesar en el análisis.

Es por ello, que se ha decidido crear un campo en la tabla el cual actuará como un semáforo, es decir, cuando en las diferentes medidas calculadas anteriormente todos los registros del cliente sean igual a 0, el semáforo tendrá el valor de 0 y si contiene algún registro con valor, entonces el semáforo será igual a 1.

El campo se denomina Control de Actividad y gracias a él será muy sencillo identificar a los clientes activos y poder descartar a los inactivos.

Para finalizar, se deben crear los campos de Nivel de Éxito de presupuestos, Nivel de Albaranes y Nivel de Tickets. A estos se les asignará un valor dependiendo de la evolución de sus respectivos documentos durante los años, como se ha explicado en el apartado de Creación del grado de satisfacción. Por lo que compararemos las columnas creadas anteriormente para darle el valor que le corresponda a cada nivel.

Tras crear los tres campos de Nivel, que serán la entrada del algoritmo de clustering, la tabla está lista para ser exportada de Qlik Sense y tendrá el siguiente formato:

Código Cliente	Q	Exito Presupuestos 2018	Exito Presupuestos 2019	Exito Presupuestos 2020	Importe Albaranes 2018	Importe Albaranes 2019	Importe Albaranes 2020	Ticket LO y AD 2019	Ticket LO y AD 2020	Control de Actividad	Nivel Exito Presupuestos	Nivel Importe Albaranes	Nivel Tickets
Totales													
0001		0,00	0,00	0,00	0,00	0,00	0,00	0	0	0	0,66	0,66	0,66
0004		0,55	0,42	0,63	11.946,31	10.765,80	11.272,84	3	2	1	0,33	0,33	1
0008		0,00	0,00	0,00	0,00	0,00	49,46	0	0	1	0,66	0,33	0,66
0020		0,00	0,00	0,00	0,00	0,00	0,00	0	0	0	0,66	0,66	0,66
0026		0,50	0,00	0,00	2.706,29	1.619,70	0,00	0	0	1	0,33	0	0,66
0044		0,00	0,00	0,00	5.776,08	1.494,95	1.852,95	0	3	1	0,33	0,33	0,33
0049		0,00	0,00	0,00	0,00	0,00	0,00	0	0	0	0,66	0,66	0,66
0050		0,50	0,33	0,00	1.177,48	4.656,59	3.487,94	1	7	1	0	0,66	0,33
0051		0,00	0,67	0,25	2.801,46	3.367,58	2.707,26	0	1	1	0,66	0,66	0,33
0055		0,00	0,00	0,30	0,00	0,00	2.755,56	0	0	1	0,33	0,33	0,66

Ilustración 27: Fragmento de la tabla lista para exportar de Qlik Sense

5.2 Python para el desarrollo de la segunda fase

La segunda fase del proyecto se ha desarrollado en Python. En este apartado, se verán las librerías utilizadas y el uso que se les ha dado en el proyecto, así como algunos tratamientos realizados.

5.2.1 Librerías utilizadas

Las librerías utilizadas para desarrollar la segunda fase del proyecto en Python son las siguientes.

5.2.1.1 Scikit-learn

Python es uno de los lenguajes más populares en la actualidad para la implementación de modelos de aprendizaje automático debido a su sencillez y potencia, pero las funciones de aprendizaje automático no se incluyen en las librerías estándar. Por lo que es necesario instalar librerías para realizar estas tareas.

Esta es una librería *open source* para el lenguaje de programación Python que implementa diferentes métodos de aprendizaje automático. Cuenta con diversos algoritmos para la implementación de los modelos más utilizados en aprendizaje automático, entre los que incluye: [4]

- Clasificación
- Regresión
- Clúster
- Reducción de la dimensionalidad

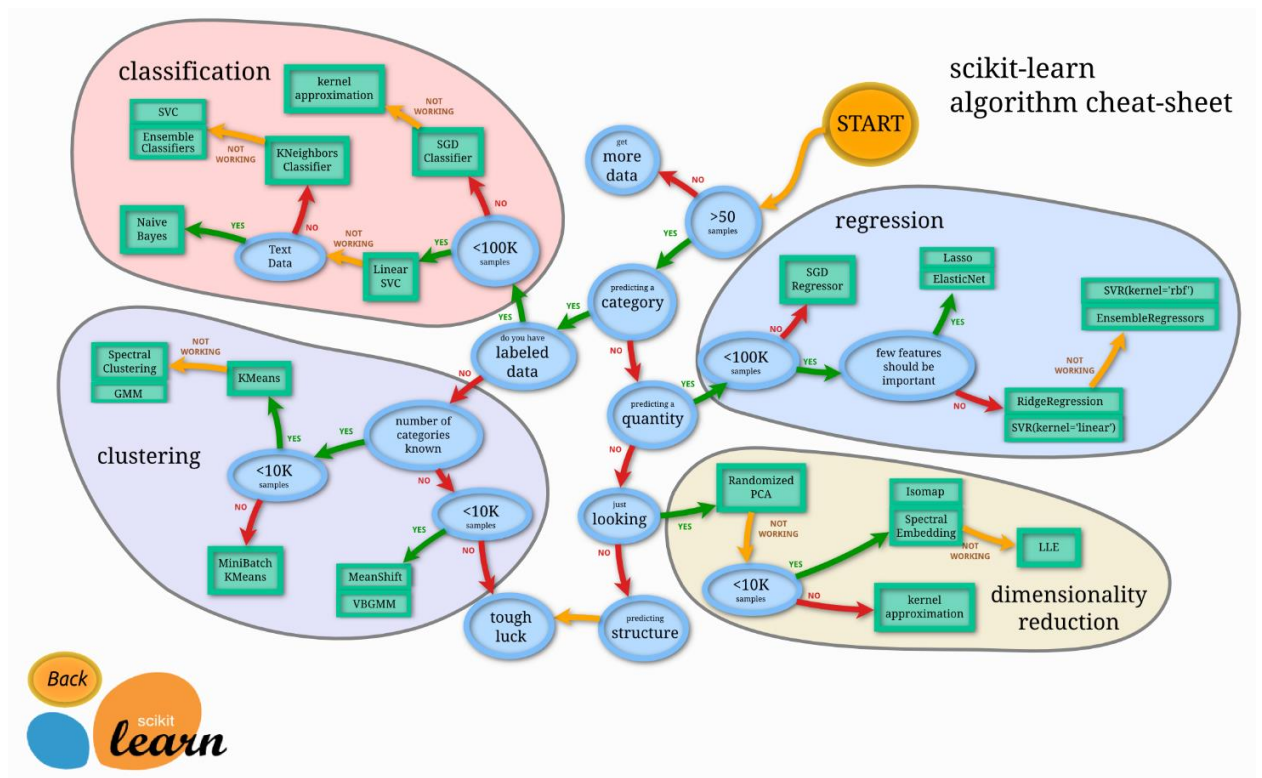


Ilustración 28: Esquema de algoritmos en scikit-learn

En la imagen superior se muestra un esquema que podemos encontrar en la propia web de scikit-learn [5]. En el que se puede observar todos los algoritmos disponibles en la librería y un proceso para seleccionar el más adecuado, dependiendo de los datos que se tengan disponibles y el análisis que se quiera llevar a cabo.

Para desarrollar el algoritmo de clustering en la segunda fase del proyecto se ha decidido utilizar la librería de Scikit-learn, en concreto el clustering de datos no etiquetados se puede realizar con el módulo `sklearn.cluster`. Dentro de este, tendremos disponible la implementación del algoritmo K-Means utilizando el método `KMeans`.

Al método `KMeans` se le deberá pasar por parámetro el número de clusters en el que deberá separar a los clientes.

Cada algoritmo de clustering viene en dos variantes: una clase que implementa el método *fit* para aprender que clusters hay sobre los datos que vamos a analizar y una función que, dados los datos a analizar, nos devuelve un array de etiquetas representadas por números enteros. Estas corresponden a los identificadores de cada clúster para cada registro de los datos pasados al algoritmo y las podemos encontrar sobre el atributo `labels_`.

5.2.1.2 Pandas

Pandas es una herramienta de manipulación y análisis de datos de código abierto rápido, potente, flexible y fácil de usar. Construida sobre el lenguaje de programación Python, está diseñada para que el trabajo con datos “relacionales o etiquetados” sea fácil e intuitivo. Su objetivo es ser el bloque de construcción fundamental de alto nivel para realizar análisis de datos prácticos del mundo real en Python. Además, tiene el objetivo más amplio de convertirse en la herramienta de análisis y manipulación de datos de código abierto más potente y flexible disponible en cualquier idioma. [6]

En el proyecto, se ha utilizado la librería de pandas para leer y escribir los archivos Excel que serán la entrada y la salida de la segunda fase.

También, se ha utilizado para eliminar filas y columnas del DataFrame inicial, y así obtener un DataFrame con únicamente los clientes activos y las tres columnas necesarias para pasárselo como entrada al algoritmo de Machine Learning. Una vez obtenido el clúster al que pertenece cada cliente, se ha utilizado la librería de pandas para insertar el DataFrame, que contiene las etiquetas de los clusters, al DataFrame inicial, que contiene los códigos de los clientes, para poder ser exportado.

5.2.1.3 Matplotlib

Matplotlib es una librería completa para crear visualizaciones estáticas, animadas e interactivas en Python. [7]

Utilizando esta librería se puede llegar a desarrollar gráficos de gran calidad con solo unas pocas líneas de código. También, se puede personalizar una gran variedad de aspectos en la visualización como los estilos de línea, las propiedades de la fuente, las propiedades de los ejes, etc. Y finalmente, es posible exportar los gráficos en diferentes formatos.

En el proyecto, se ha utilizado la sublibrería pyplot, ya que solo se utilizará para realizar alguna visualización sencilla y esta sublibrería está diseñada principalmente para gráficos interactivos y casos simples de generación de gráficos, así será suficiente para la segunda fase del proyecto.

5.2.2 Tratamiento de los datos

Si recordamos el tratamiento y la transformación de los datos que realizamos en Qlik Sense, se terminan creando cuatro campos nuevos, los cuales adquieren un valor u otro dependiendo de sus anteriores registros en la visualización de tabla. El campo de Control de Actividad se utilizará para dividir el dataset, y los tres campos de Nivel se le pasarán como conjunto de datos a analizar al algoritmo K-Means.

Así pues, el DataFrame inicial será el siguiente:

Índice	idigo Clie	Presupuesto	resupuest	resupuest	orte Albaranes	orte Albaranes	orte Albaranes	t LO y AD	t LO y AD	ntrol de Activic	ivel Exito Presupuestc	livel Importe Albarane	Nivel Tickets
0	1	0.000	0.000	0.000	0.000	0.000	0.000	0	0	0	0.660	0.660	0.660
1	4	0.545	0.417	0.625	11946.310	10765.800	11272.840	3	2	1	0.330	0.330	1.000
2	8	0.000	0.000	0.000	0.000	0.000	49.460	0	0	1	0.660	0.330	0.660
3	20	0.000	0.000	0.000	0.000	0.000	0.000	0	0	0	0.660	0.660	0.660
4	26	0.500	0.000	0.000	2706.290	1619.700	0.000	0	0	1	0.330	0.000	0.660
5	44	0.800	0.000	0.000	5776.080	1494.950	1852.950	0	3	1	0.330	0.330	0.330
6	49	0.000	0.000	0.000	0.000	0.000	0.000	0	0	0	0.660	0.660	0.660

Ilustración 29: Fragmento del DataFrame inicial

Si se observan los clientes activos e inactivos dentro del DataFrame, basándonos en el campo de Control de Actividad, podemos ver que el número de clientes activos es de 2826 mientras que el número de clientes inactivos es de 718.

Ya que solo se debe analizar a los clientes que estén activos, se necesita separar a los clientes activos y los inactivos. Para ello, se seleccionan solo los registros que tienen el valor de 1 en el campo de Control de Actividad.

Una vez seleccionados los clientes activos, se deben eliminar todas las columnas menos las tres últimas mediante el método *drop* de pandas. De esta forma, se tendrá el DataFrame preparado para que sea la entrada del algoritmo K-Means.

Tras realizar estas operaciones el DataFrame denominado *data_activos* contendrá 2826 filas y 3 columnas, y quedará de la siguiente forma:

Índice	Nivel Exito Presupuestos	Nivel Importe Albaranes	Nivel Tickets
0	0.330	0.330	1.000
1	0.660	0.330	0.660
2	0.330	0.000	0.660
3	0.330	0.330	0.330
4	0.000	0.660	0.330
5	0.660	0.660	0.330
6	0.330	0.330	0.660

Ilustración 30: DataFrame final al que aplicar el algoritmo

5.2.3 Aplicando el método del codo

Una vez obtenido el dataset que se quiere utilizar para desarrollar el análisis, es hora de pasar a realizar el clustering. Como se va a utilizar el algoritmo de K-Means para obtener los clusters de los datos de entrada, se debe seleccionar primero el número de clusters óptimo, para ello se utilizará la técnica del codo.

Primero, se creará una lista vacía en la cual se almacenarán los valores de la inercia obtenidos en cada ejecución del algoritmo de K-Means para diferente número de clusters, esta lista se denominará *scores*.

Tras esto, se crea un bucle el cual determinará el número de veces que se ejecuta el algoritmo y le aplicará en cada iteración un clúster más para el valor de K. En este caso, se ha decidido que con 10 iteraciones será suficiente para determinar el número correcto de clusters a utilizar.

Dentro del bucle, inicializamos el algoritmo de KMeans con el valor de K que se le asigne en cada iteración del bucle. Tras esto, se aplica el método *fit* pasándole por parámetro el data set de entrada. Finalmente, se añadirá el valor de la inercia obtenido en cada ejecución a la lista de *scores*.

```
for k in range(1,11):
    kmeans = KMeans(n_clusters = k)
    kmeans.fit(data_activos)
    scores.append(kmeans.inertia_)
```

Ilustración 31: Implementación método del codo

Como se ha comentado anteriormente, el número óptimo de clúster se debe seleccionar de forma visual, así que se tiene que mostrar el gráfico obtenido para que el usuario pueda seleccionar el número de clusters correcto.

Para ello, mediante la librería matplotlib y su método *plot*, se creará un gráfico en el que el eje de las X contendrá el número de clúster seleccionados y el eje de las Y, contendrá el valor de la inercia para cada número de clusters.

Como resultado, se obtendrá un gráfico similar al siguiente:

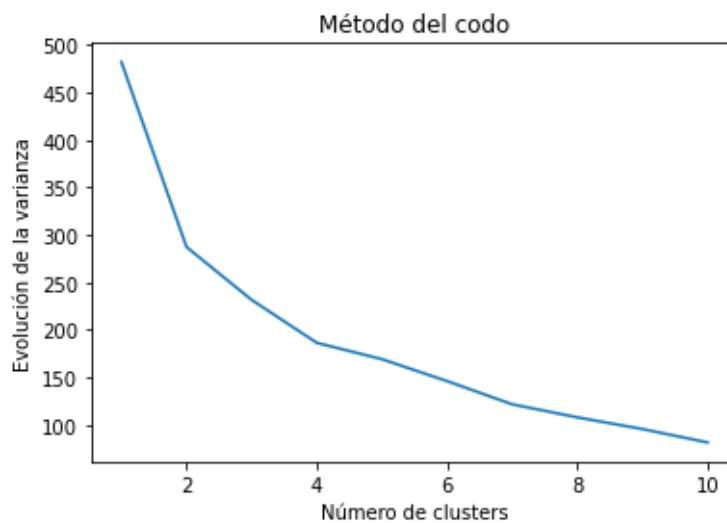


Ilustración 32: Visualización del método del codo

Como se puede observar el número óptimo de agrupaciones, es decir el “codo”, en este caso estaría situado en el número de clusters igual a 4.

5.2.4 Aplicando el algoritmo de K-Means

Una vez se conoce el número óptimo de clusters, se aplicará el algoritmo K-Means con el valor óptimo para el parámetro K, que como se ha visto será 4.

Se implementará el algoritmo como el apartado anterior, pero esta vez no se debe realizar ningún bucle sino que simplemente se le pasará como parámetro el valor 4.

Finalmente, se deberá almacenar el valor del clúster que pertenece a cada cliente en una variable. Estos registros quedarán guardados como etiquetas en el atributo *labels_* dentro del objeto KMeans.

Esta variable, que será una lista, se deberá concatenar al DataFrame inicial con los clientes activos para finalmente ser exportada en formato Excel. De esta forma, se podrá obtener el código de cliente que tiene documentado SEMIC en su ERP con el clúster al que pertenece para poder realizar un análisis interno.

5.2.5 Elemento central de cada clúster

También será importante para el usuario conocer el valor de cada centroide de los clusters, de esta forma se podrá identificar las características de los clientes encontraremos en cada grupo, y con esto saber que grupo pertenece a clientes más satisfechos o menos satisfechos. Se pueden recuperar los centros de cada clúster mediante el método *cluster_centers_*.

Para poder analizar correctamente los valores centrales de cada clúster, se debe saber que el valor máximo será el 1 y el valor mínimo será el 0. Si el valor tiende hacia el 1, será síntoma de satisfacción mientras que si tiende hacia el 0 será síntoma de insatisfacción.

Índice	Nivel Exito Presupuestos	Nivel Importe Albaranes	Nivel Tickets
0	0.666	0.801	0.330
1	0.282	0.289	0.528
2	0.649	0.723	0.707
3	0.678	0.226	0.536

Ilustración 33: Valor de los centroides de cada clúster

En la tabla anterior, se puede interpretar que en el clúster 2 se encuentran los clientes más satisfechos, ya que tienen un buen nivel de éxito en presupuestos, un excelente nivel de importe en albaranes y un buen nivel en ticket. Por otro lado,

podemos observar que en el clúster 1 se sitúan los clientes con menor satisfacción, ya que tienen un muy mal nivel de éxito en presupuesto e importe de albaranes, y el nivel de tickets se sitúa en la parte intermedia. Por último, también identificamos en el clúster 0 a clientes que trabajan bastante con la empresa, es decir, que tienen un valor alto en éxito de presupuestos y en importe, pero generan un gran volumen de incidencias.

Estos valores, serán exportados también en formato de Excel y se cargarán en la visualización para que el usuario final sea capaz de interpretar la situación de satisfacción en la que se encuentran los clientes de cada clúster.

5.3 Visualización

Como se ha visto en el apartado de Datos Añadidos en la sección de Diseño, para poder realizar las visualizaciones que podrá utilizar el usuario final de la herramienta, se deben cargar nuevas tablas y reformular el modelo de datos. Así que, para generar las visualizaciones se partirá del modelo mostrado en ese apartado.

5.3.1 Modificaciones en la carga de datos

En este apartado no se entrará a explicar tan al detalle cómo se han cargado las tablas, ya que se realiza de forma similar a la primera fase, pero se comentarán los detalles más significativos.

- Tabla de Documentos:

En la tercera fase, en vez de cargar las tablas de las cabeceras de los documentos por separado se cargarán concatenadas entre sí, formando una única tabla. Todas compartirán el campo *id_linea*, el cual será la relación con sus respectivas líneas, y también el campo *id_cliente*, que será la relación con la tabla de clientes.

Para ser capaces de diferenciar el tipo de documento, se ha creado un campo denominado *clase_doc*, al cual se le asignará un valor dependiendo del tipo de documento que sea. Para los presupuestos tendrá el valor “PR”, para los albaranes tendrá el valor “AV” y para los pedidos tendrá el valor “PV”. De esta forma, en la visualización se podrá filtrar por el documento del que se deseen mostrar sus datos.

- Tabla de Líneas:

En la última fase del proyecto, se deberán cargar las líneas de los tres documentos nombrados anteriormente para poder tener la máxima información posible sobre estos, a diferencia de la primera fase que solo se cargaban las líneas de los albaranes.

De la misma forma que las cabeceras de los documentos, las líneas también se deberán concatenar entre sí. De esta forma, se reduce el número de tablas a cargar, ya que con una única tabla de artículos podremos conocer el detalle de estos en las líneas de cada documento. Si no se concatenaran las líneas, se debería cargar la tabla de artículos para cada documento y así evitar las referencias circulares.

Las líneas de cada documento compartirán el campo *id_linea*, el cual será la relación con sus respectivas cabeceras del documentos, y el campo *id_articulo*, el cual será la relación con la tabla de artículos.

En cuanto al campo de *clase_doc* en las líneas no será necesario, ya que al estar relacionada cada línea con su documento, actuará de filtro también para estas líneas. Es decir, si se filtra por *clase_doc* igual a “AV” nos mostrará solo los documentos que pertenezcan a los albaranes y por lo tanto, también solo sus líneas.

5.3.2 Organización de la visualización

En este apartado, se verá como se ha decidido estructurar y realizar las visualizaciones. Aunque se tratará de una propuesta que el usuario final de negocio tendrá que validar, ya que será este quien consuma la herramienta.

Las visualizaciones en Qlik Sense se pueden dividir por hojas, así que se ha decidido crear una hoja para cada documento y otra para los clientes. Es decir, una hoja para los albaranes, otra para los pedidos, otra para los presupuestos, otra para los tickets y finalmente, otra para los clientes.

- Filtros de Clúster:

En todas las hojas de visualizaciones estarán disponibles filtros para seleccionar el clúster del que se quieren consultar los datos y a su lado una tabla mostrando el valor de los “centroides” de cada clúster para que el usuario pueda reconocer que tipo de cliente se encuentra en cada clúster.

Un ejemplo en la visualización podría ser la siguiente:

Cluster		Puntos Centrales del Cluster			
Cluster		Nivel Éxito Presupuestos	Nivel Importe Albaranes	Nivel Ticket	
0					
1	0	0,666	0,801	0,33	
2	1	0,282	0,289	0,528	
3	2	0,649	0,723	0,707	
	3	0,678	0,226	0,536	

Ilustración 34: Filtro y tabla de centroides de los clusters en la visualización

• Visualización de Clientes:

En esta hoja de visualizaciones, se podrá encontrar todas las visualizaciones relacionadas con los clientes. Como por ejemplo, a qué provincia pertenece los clientes de cada clúster, a que población pertenecen o que métodos de pago utilizan.

Un ejemplo podría ser el siguiente:

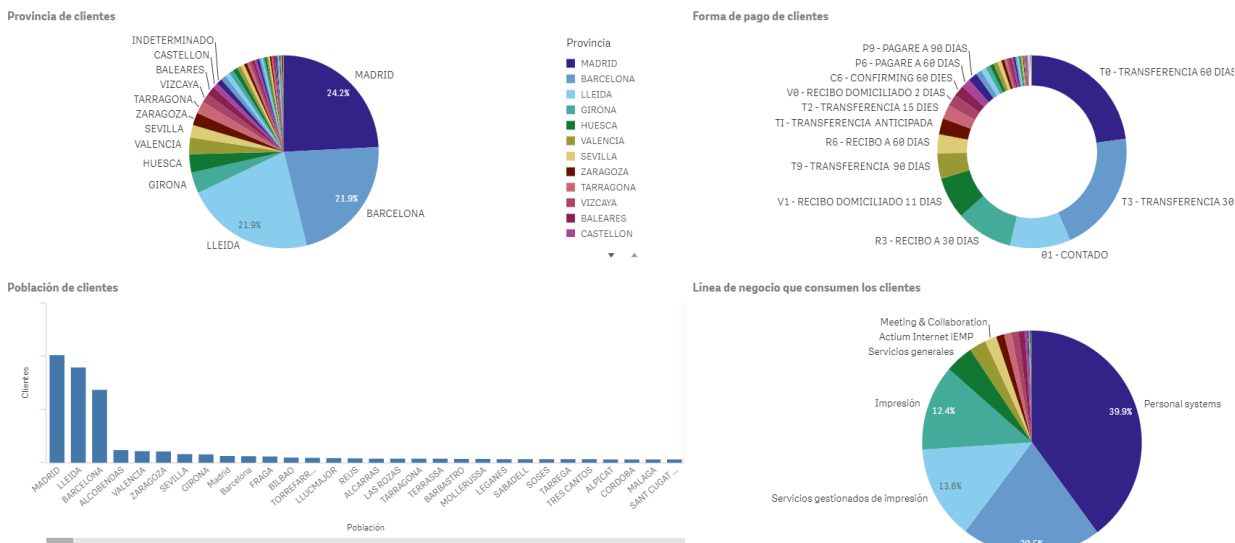


Ilustración 35: Ejemplo de visualización de los datos de Clientes

Bajo estas visualizaciones se mostrará un detalle en formato tabla donde se podrá conocer más profundamente a los clientes, dando informaciones como el nombre del cliente, el N.I.F. del cliente, su localización, su código de cliente, etc.

- Visualización de Documentos

Esta visualización será similar en las hojas de Albaranes, Pedidos y Presupuestos ya que contienen una información semejante. Para dar algo de visión sobre cómo podría ser la visualización y no repetir el proceso tres veces, se va a mostrar con el ejemplo de los albaranes.

En la hoja de visualizaciones de Albaranes se podrán apreciar los datos relacionados a los albaranes que tienen los clientes de cada clúster. Como por ejemplo, la línea de negocio de los albaranes, los artículos que contienen los albaranes, el agente asignado o el responsable de la operación entre otros.

Un posible ejemplo de visualización podría ser la siguiente:

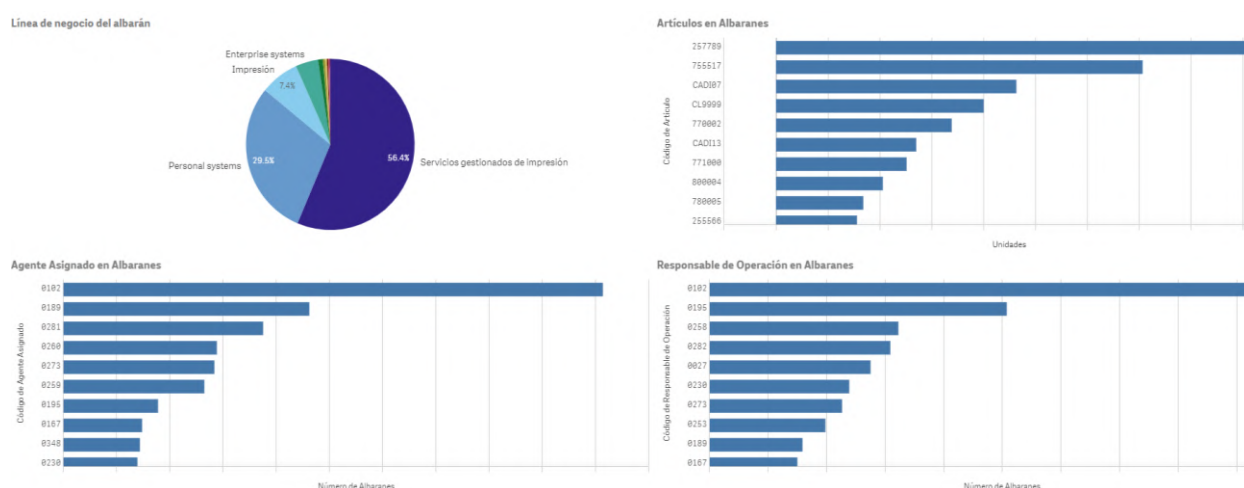


Ilustración 36: Ejemplo de visualización de los datos de Albaranes

Bajo las visualizaciones también se mostrará un detalle de los albaranes en formato tabla, dando a conocer el identificador del albarán, el cliente, la fecha, la línea de negocio, agente asignado, responsable de operación, artículos, unidades, importe, etc.

- Visualización de Tickets:

En esta hoja de visualización se podrá observar la información relacionada con los tickets generados por el cliente durante el proceso de una venta. Por ejemplo, se podrá visualizar el tipo del ticket, la categoría del ticket, los errores por los que se ha generado el ticket o el responsable del ticket entre otros.

Un ejemplo de visualización podría ser el siguiente:

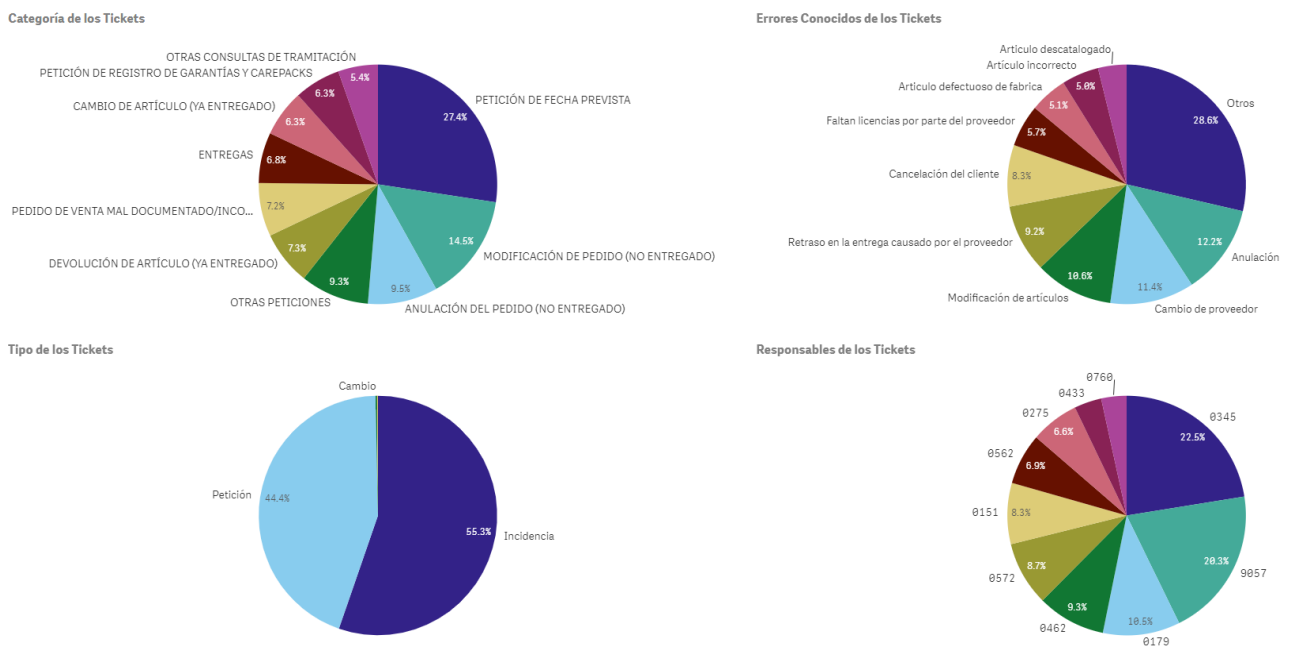


Ilustración 37: Ejemplo de visualización de los datos de Tickets

Una vez más, bajo los gráficos se podrá apreciar un detalle dando información más precisa sobre los tickets, como su identificador, su estado, su fecha de alta, su fecha de resolución, etc.

6 Casos de uso

En esta sección, se mostrarán algunos de los posibles casos de uso que el usuario final de la herramienta podría realizar.

6.1 Detección de artículos que generan satisfacción o insatisfacción

Un posible caso de uso podría ser analizar los diferentes artículos que nos devuelve la herramienta al filtrar por el clúster al que pertenecen los clientes con mayor nivel de satisfacción y por el clúster al que pertenecen los clientes con menor nivel de satisfacción.

Por ejemplo, si se observan los artículos de los albaranes pertenecientes a los clientes del clúster número 2, estos serán los clientes más satisfechos, se puede observar que tienen una mayor presencia los siguientes códigos de artículos.

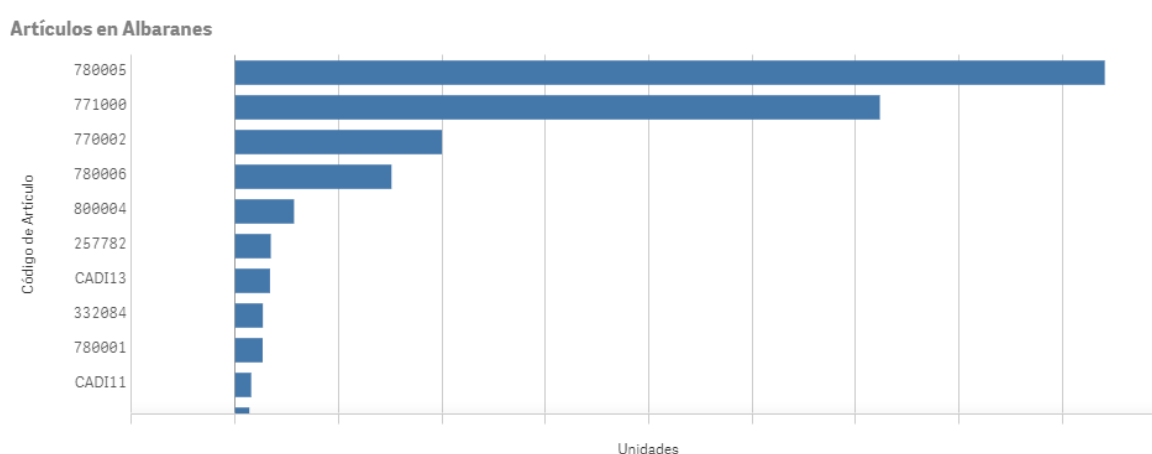


Ilustración 38: Artículos consumidos por clientes con mayor nivel de satisfacción

Por otro lado, si se observan los artículos de los albaranes pertenecientes a los clientes que pertenecen al clúster número 1, es decir, los clientes con menor nivel de satisfacción, se puede ver que tienen una mayor presencia los artículos representados en el siguiente gráfico.

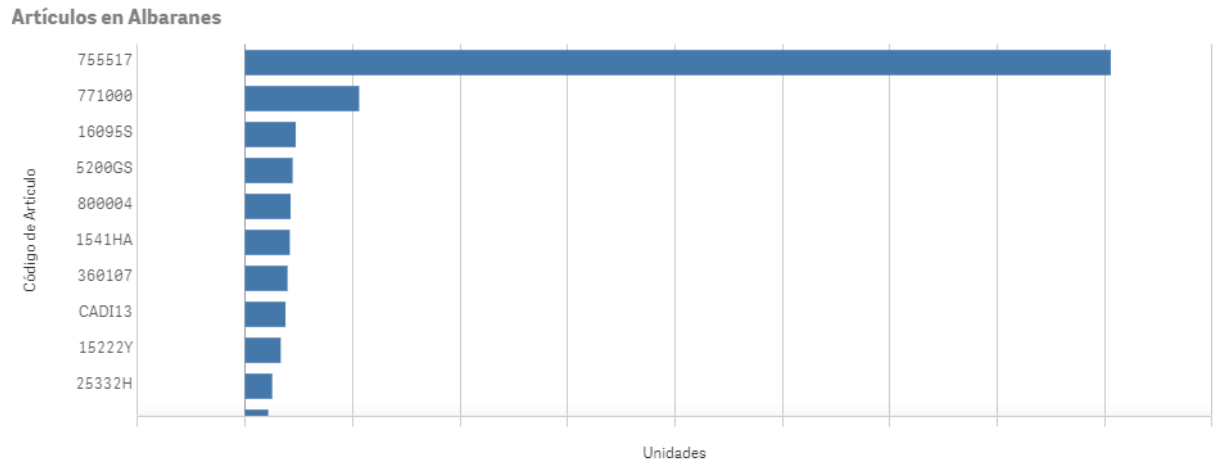


Ilustración 39: Artículos consumidos por clientes con menor nivel de satisfacción

Como se puede contemplar, los clientes que aparecen agrupados en el clúster de menor nivel de satisfacción, consumen con una gran diferencia el artículo con código 755517. Pero, si observamos la misma gráfica para los clientes con mayor nivel de satisfacción, se puede observar que este artículo no aparece en el gráfico y por lo tanto, no será un producto que consuman a gran escala estos clientes.

Para concluir este caso de uso, se podría decir que el artículo con código 755517 puede estar generando insatisfacción entre los clientes que lo consumen, ya que es un artículo común entre los clientes con menor nivel de satisfacción, y por lo tanto ser una posible causa de su insatisfacción.

A partir de aquí, sería necesario que los expertos del negocio realicen un análisis para comprobar y mejorar la satisfacción de los clientes con este artículo.

Además de poder detectar productos que producen insatisfacción, este caso de uso se puede utilizar también para observar que productos generan una mayor satisfacción y así fomentar su venta.

Este sería el caso del artículo con código 780005, el cual aparece como artículo más consumido entre los clientes con mayor nivel de satisfacción y no aparece entre los principales artículos consumidos por los clientes con menos nivel de satisfacción.

6.2 Provincia de los clientes

Otro caso de uso, podría ser analizar las diferentes provincias o poblaciones a las que pertenecen los clientes con mayor o con menor nivel de satisfacción. Pudiendo dar visión sobre si se está actuando de forma incorrecta sobre algún territorio.

Si se observan las provincias a las que pertenecen los clientes con mayor nivel de satisfacción, se podrán encontrar las siguientes:

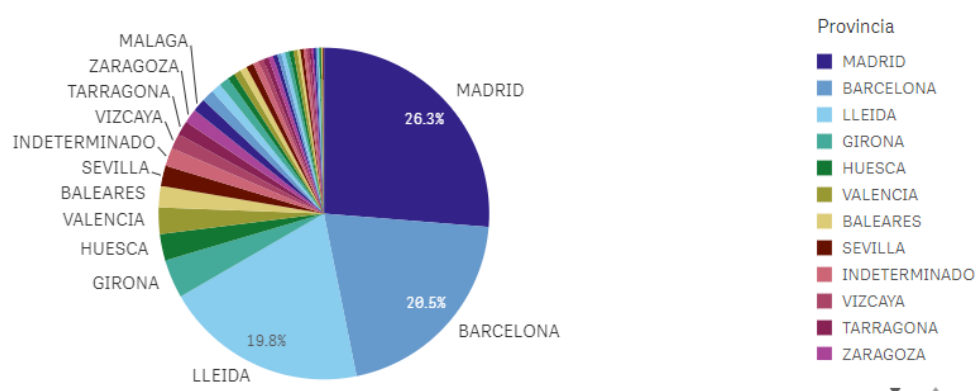


Ilustración 40: Provincias de los clientes con mayor nivel de satisfacción

Como se puede ver, el 26.3% de los clientes con mayor nivel de satisfacción pertenecen a la provincia de Madrid, seguido de la provincia de Barcelona con un 20.5% y la provincia de Lleida con un 19.8%.

Mientras que, si observamos las provincias a las que pertenecen los clientes con menos nivel de satisfacción, se puede observar el siguiente gráfico:

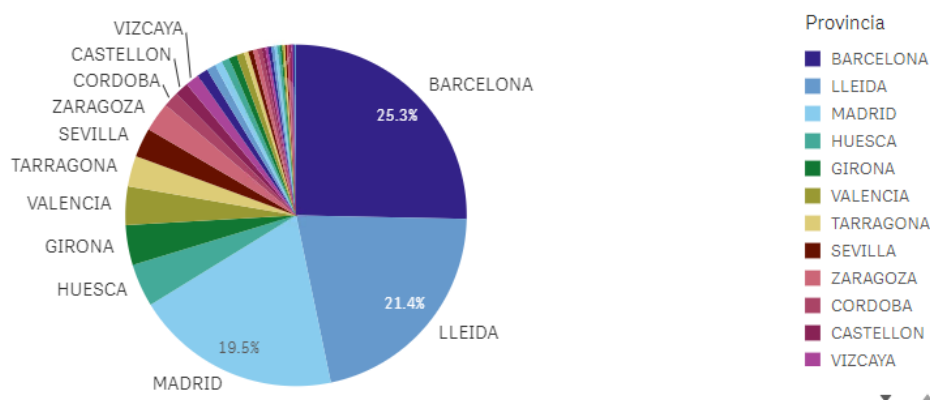


Ilustración 41: Provincias de los clientes con menor nivel de satisfacción

En esta, se puede observar que el 25.3% de los clientes con menor nivel de satisfacción pertenecen a la provincia de Barcelona, seguido de la provincia de Lleida con un 21.4% y de la provincia de Madrid con un 19.5%.

Con los datos de entrada utilizados de la empresa SEMIC, este caso de uso no muestra unos resultados relevantes. Se puede observar que los clientes que pertenecen a la provincia de Madrid tienden algo más hacia un buen nivel de satisfacción, pero no se pueden apreciar grandes diferencias entre las provincias a las que pertenecen los clientes con mayor y menor nivel de satisfacción.

En el caso de que se realizara el análisis en otro periodo o en una empresa diferente, podría aportar una buena información a la hora de conocer en que territorios la empresa tiene una mejor satisfacción del cliente y en cuales una menor satisfacción para proceder a mejorar la situación.

7 Análisis de Resultados

El objetivo de este trabajo ha sido proveer a la empresa SEMIC de una herramienta mediante la cual, los responsables de negocio sean capaces de encontrar anomalías o posibles mejoras durante el proceso de la venta al cliente. Estas se verán reflejadas dependiendo del grado de satisfacción del cliente.

Tras filtrar a los clientes que se desean cargar y, apoyándonos en la métrica, creada a partir de la evolución de ciertos documentos, denominada satisfacción del cliente, se logra agrupar a todos los clientes activos en función de la similitud que tienen con las condiciones utilizadas para definir la satisfacción.

Una vez identificado a que agrupación pertenecen los clientes con un mayor nivel de satisfacción y los que tienen un menor nivel de satisfacción, se procede a visualizar la información recuperada.

Para poder ofrecer una información más clara al usuario, se crean diferentes gráficos que le proporcionan una visualización interactiva, teniendo la oportunidad de seleccionar los datos mediante filtros. También, se tiene la posibilidad de ampliar los documentos en los que se desea consultar la información de forma sencilla añadiéndolos al modelo de datos y si, con el paso del tiempo, se incorporan nuevas características o atributos a los clientes o al resto de tablas cargadas, la herramienta no se verá afectada en su funcionamiento.

Esta herramienta se ha diseñado para que los responsables de cada departamento puedan conocer las diferentes características que comparten los clientes con un mayor y menor grado de satisfacción, y tras esto sean capaces de identificar las posibles causas que puedan generar que este valor de satisfacción aumente o disminuya en el cliente.

Como se ha podido observar en el primer caso de uso, se ha logrado obtener algún posible motivo por los que los clientes de SEMIC pueden disminuir su satisfacción con la empresa, en este caso se trata de un producto que podría estar generando insatisfacción en el cliente. También es cierto, que se debe realizar un análisis más profundo a nivel de negocio para determinar si se trata de un producto problemático o defectuoso y modificar su proceso de venta.

Para concluir, se puede decir que a través de esta herramienta los responsables de la empresa podrán obtener, a partir del grado de satisfacción de los clientes, una gran variedad de casuísticas con las que analizar y obtener conclusiones para mejorar los procesos de negocio y problemas futuros. La función principal de la herramienta, como se ha comentado, es ayudar a identificar posibles causas por las que se puede generar insatisfacción y se ha completado exitosamente.

8 Bibliografía

- [1] R. Moya, «Jarroba,» [En línea]. Available: <https://jarroba.com/seleccion-del-numero-optimo-clusters/>.
- [2] Q. Sense, «Qlik Help,» [En línea]. Available: https://help.qlik.com/es-ES/sense/February2021/Content/Sense_Helpsites/Home.htm.
- [3] Wikipedia, «Wikipedia: MyODBC,» 2020. [En línea]. Available: <https://es.wikipedia.org/wiki/MyODBC>.
- [4] «Scikit-learn,» [En línea]. Available: https://scikit-learn.org/stable/tutorial/machine_learning_map/index.html.
- [5] Scikit-learn, «Choosing the right estimator,» [En línea]. Available: https://scikit-learn.org/stable/tutorial/machine_learning_map/index.html.
- [6] pandas, «pandas,» [En línea]. Available: <https://pandas.pydata.org/>.
- [7] Matplotlib, «Matplotlib,» [En línea]. Available: <https://matplotlib.org/>.